CRANFIELD UNIVERSITY

DARREN WILLIAM ANSELL

ANTENNA PERFORMANCE OPTIMISATION USING EVOLUTIONARY ALGORITHMS

DEPARTMENT OF AEROSPACE POWER AND SENSORS

Ph.D. THESIS

CRANFIELD UNIVERSITY

DEPARTMENT OF AEROSPACE POWER AND SENSORS
Ph.D. Thesis

Academic Year 2003-2004

Darren William Ansell

Antenna Performance Optimisation Using Evolutionary Algorithms

Project Supervisor: Dr E.J.Hughes.
Presented: 18th June 2004

Abstract

This thesis investigates the novel idea of using evolutionary algorithms to optimise control and design aspects of active array antenna systems. Active arrays differ from most mechanically scanned antennas in that they offer the ability to control the shape of their radiation pattern. As active arrays consist of a multiplicity of transmit and receive modules (TRMs), the task of optimally controlling them in order to generate a desired radiation pattern becomes difficult. The control problem is especially true of conformal (non-planar) array antennas that require additional phase control to achieve good radiation pattern performance.

This thesis describes a number of significant advances in the optimisation of array antenna performance. Firstly a genetic algorithm (GA) is shown to be effective at optimising both planar and conformal antenna performance. A number of examples are used to illustrate and promote the basic optimisation concept. Secondly, in this thesis the techniques are advanced to apply multiobjective evolutionary optimisation algorithms to array performance optimisation. It is shown that Evolutionary Algorithms allow users to simultaneously optimise many aspects of array performance without the need to fine-tune a large number of weights. The multiple-objective analysis methods shown demonstrate the advantages to be gained by holding knowledge of the Pareto optimal solution set.

Thirdly, this thesis examines the problems of optimising the design of large (many element) array antennas. Larger arrays are often divided into smaller sub-arrays for manufacturing reasons and to promote formation of difference beam patterns for monopulse operation. In the past, the partitioning has largely been left to trial-and-error or simple randomisation techniques. This thesis describes a new and novel approach for optimally subdividing both planar and conformal array antennas as well as improving gain patterns in a single optimisation process. This approach contains a new method of partitioning array antennas, inspired from a biological process and is also presented and optimised using evolutionary algorithms. Additionally, the technique can be applied to any size or shape of array antenna, with the processing load dependent on the number of subarrays, rather than the number of elements.

Finally, the success of these new techniques is demonstrated by presenting a range of performance optimised examples of planar and conformal array antenna installations including examples of optimally evolved subarray partitions.

Keywords: Genetic Algorithm, AESA, phased array, optimisation, synthesis, multiobjective, subarray, conformal, beam pattern, partitioning

Acknowledgements

I would like to thank Keith Henderson and Harry Finn of BAE SYSTEMS for introducing me to radar and antenna technology and for their encouragement and unreserved support over the last four years. I would especially like to thank Dr Evan Hughes of Cranfield University for introducing me to the science of modern optimisation, supervising my PhD work and for making this work possible.

Glossary of Terms

AESA	Active Electronically Scanned Array
ANN	Artificial Neural Network
E-Scan	Electronically Scanned
EA	Evolutionary Algorithm
EP	Evolutionary Programming
ES	Evolutionary Strategy
ESM	Electronic Support Measures
EW	Electronic Warfare
FFT	Fast Fourier Transform
GA	Genetic Algorithm
IEE	Institution of Electrical Engineers
IFF	Identify Friend or Foe
MOEA	Multi-Objective Evolutionary Algorithm.
MOGA	Multi-Objective Genetic Algorithm
MoM	Method of Moments
NPGA	Horn, Nafploitis, and Goldberg's Niched Pareto Genetic Algorithm.
NSGA	Srinivas and Deb's Non-Dominated Sorting Genetic Algorithm.
PAES	Knowles and Corne's Pareto Archived Evolution Strategy
PESA	Pareto Envelope-Based Selection Algorithm
$\mathbf{P_{cross}}$	Probability of Crossover
$\mathbf{P_{mut}}$	Probability of Mutation
PSO	Particle Swarm Optimisation
SGA	Simple (or Standard) Genetic Algorithm
SPEA(II)	Zitler's Strength Pareto Evolutionary Algorithm (II).
TRM	Transmit Receive Module
VEGA	Schaffer's Vector Evaluated Genetic Algorithm

Contents

Figures

Tables

CHAPTER 1
Introduction

1. Introduction

1.1. Active Electronically Scanned Antenna Arrays

Active phased array antennas [1] are rapidly replacing mechanically scanned antennas as the system of choice in land, air and sea-based systems. This popularity is due to their versatility, reliability and the gradually reducing cost of their component parts. Active arrays are being used primarily in radar systems but have also found application in communication, navigation and electronic warfare systems.

Active arrays consist of a multiplicity of transmit receive modules (TRMs) [2] arranged in either a linear, planar or conformal arrangement. The single most attractive benefit of active array antennas is the ability to rapidly change and control the shape of the antenna radiation pattern. This capability brings many operational benefits such as the ability to adapt the pattern as the specific radio frequency task, threat or environment changes. It also allows the antenna to scan rapidly across its entire field of view. Such scans need not be constrained to raster or bar-type scans as with mechanically scanned antennas, but instead can rapidly switch beams between extremes of angular scan if required.

Each application of active array technology brings its own challenges and quite different operating requirements. For example, radar functions may require variable gain, low sidelobes and narrow beamwidth while communication systems may need a wide beamwidth and fixed gain.

1.2. Control of Active Arrays

While active arrays offer far more versatility than parabolic, mechanically scanned antennas, they are much more difficult to build and to control. TRM technology is improving all the time, but the achievable transmit-power level per module is still only of the order of a few watts when transmitting in X-Band at 8-12 GHz. Therefore to achieve high transmit power levels, as many as several thousand TRMs are required.

RF emissions generated using recent TRM designs can be digitally controlled in both amplitude and phase. If several thousand TRMs are used to build an array antenna, the set of amplitude and phase values known as the excitation set can become large (although it is not strictly necessary to actively control every element in the array). The question of how to choose values for the excitation set forms a difficult optimisation problem.

Generally speaking, the phase values are set to steer the mainlobe in a desired direction, while the amplitude values are used to 'shape' the pattern. Therefore by using the correct amplitude and phase weightings, the mainlobe can be steered in the correct direction, and the pattern shaped in some manner to reduce sidelobes or influence the beamwidth. It is also possible to shape beams by varying amplitude alone, phase alone, by varying the array geometry or the choice of radiating elements.

The scale of the control problem increases with the array size, and so it common practice for manufacturers to 'subdivide' or partition large arrays into a number of smaller arrays in order to simplify their control and manufacture. While this partitioning reduces the scale of the problem into more manageable pieces, it is often at the expense of array performance and is discussed in more detail below.

A further dimension is added to the control problem when conformal arrays are considered. Conformal arrays are best described as non-planar arrays (i.e. they conform to some shaped surface, often the curved shape of an aircraft fuselage, missile or some other host vehicle). They are more difficult to control than planar arrays. The fact that the TRMs do not radiate in the same direction can introduce polarisation errors. The elements can be rotated about their axis and also tilted and so necessitate a higher degree of phase control in order to generate satisfactory radiation patterns. As with large planar arrays, conformal arrays are more difficult to control efficiently as they increase in size. Conformal arrays typically have higher sidelobes than planar arrays due to their non-linear arrangement of TRMs [3].

1.3. Hypothesis

The hypothesis of this thesis is that modern optimisation techniques can be used to provide design solutions for both planar and conformal antenna arrays, whilst maximising the performance of the array radiation patterns. This thesis aims to investigate this hypothesis by:

1. Examining traditional approaches to array design and control,
2. Examining the current range of modern optimisation methods,
3. Modelling and optimising a range of antenna configurations.

1.4. Classical Methods of Pattern Synthesis

Classical amplitude distributions such as the 'Taylor' [4] or 'Dolph-Chebyshev' [5,6] have often been chosen in the past as effective means of achieving radiation pattern objectives such as narrowest possible beam width for a given sidelobe level. In theory they give very good performance on both planar and conformal arrays, but they rarely achieve the theoretically possible performance due to amplitude and phase control

errors with the TRMs. As the control of array excitation is quantised, Taylor and Dolph-Chebyshev amplitude weightings can only be approximated in practice.

There are also practical issues in the implementation of Dolph-Chebyshev weightings as they require high currents at the edges of the array causing heat concentrations and affecting the efficiency of cooling systems, although this is less of a problem on small arrays.

There are other classical means of pattern synthesis such as the Schelkunov form [7] or Woodward synthesis [8]. These techniques operate under a set of assumptions such as the array being perfect which is of course not true in real hardware. The Woodward technique for example, works to shape specific parts of the radiation pattern and does not control sidelobes elsewhere in the pattern. These analytical approaches were conceived around the time that practical array antennas were first being considered (1930-40's) and computer processing was not available. Analytical techniques tend to assume that the array pattern is a product of an element pattern (i.e. the radiation pattern of a single radiating element) and an array factor (geometry dependent function). The pattern of an element in an array is not the same as the pattern of the same element used in isolation. Exciting one element in the array produces radiation from that element and additional radiation from all other elements in the array, because of the currents induced in them by the excited radiator. This effect is known as mutual coupling [9].

Therefore, in an installed active array antenna, every TRM in the array will produce a slightly different gain pattern and these should be taken into account when choosing optimal phase and amplitude weightings. In addition, it is well known that effects such as flashlobes attributed to radomes [10] further distort the beam pattern from the ideal. Changing the phase of radiating elements can help to correct for these effects, but the techniques above tend to assume that phase is fixed.

When the classical amplitude distributions are applied to conformal arrays they can perform well but it must be remembered that they only provide one type of radiation pattern (i.e. they solve one problem at a time such as providing a high gain, low sidelobe pattern). It is shown in this thesis, that other amplitude weightings can be found that offer improved mainlobe gain levels at the compromise of some far-out sidelobe levels (see Chapter 6).

As additional functionality is added to systems, many more diverse radiation patterns are required and new methods for determining excitation sets become necessary. Classical approaches provide good amplitude weightings (optimal in some cases) for radiation patterns with low sidelobes and narrow beam widths. The classical patterns are designed to be optimal given a wide set of assumptions, such as isotropically radiating elements or infinite arrays. Gradient based optimisers such as Newton's technique have been used but they are time consuming to calculate by hand if knowledge of many different patterns are required, perhaps for multifunction use of an antenna array.

As more sophisticated antenna patterns are required to enhance and exploit the capabilities of arrays (such as multiple beam generation – see Chapter 10 which contains case studies) new methods of finding optimal excitation weightings are needed.

1.5. Classical Optimisation Techniques

Finding optimally performing excitation sets is a difficult problem due to the enormous size of the solution search space. To put the scale of this task into context, let us consider a 1000 element active array antenna suitable for radar applications. As discussed above, in order to create a low sidelobe beam at a desired azimuth and elevation steering angle, each element in the array requires a certain phase value and amplitude weighting. Assuming that a three-bit phase shifter controls the phase setting, and the amplitude weighting is controlled by a five-bit variable gain control, we then require an 8000-bit binary string to encode a single excitation set. This binary string results in a total search space containing $2^{8000} = 1.738 \times 10^{2408}$ possible excitation sets.

Computing an approximation to the radiation pattern performance is computationally intensive and can take several seconds for evaluation of a single – first order approximation. Given this enormous decision space, and the computation time to evaluate a single pattern, an exhaustive search for optimal excitation sets is impractical.

Unless a closed form solution for the problem can be calculated, we have no other choice but to use some kind of optimisation tool, implemented as a computer algorithm in order to find acceptable solutions for the problem. The success or otherwise of the optimiser will be determined by the strength of the final solutions it provides and the timely manner in which it provides them.

A number of the more common classical optimisation techniques used in antenna optimisation are described below, together with a discussion of their performance (given in Section 1.5.6).

1.5.1. Simple and Metropolis Monte Carlo Simulations

One of the simplest methods of optimisation is the Monte Carlo simulation. The method is generally attributed to von Neumann and Ulam, who around 1946-47 developed the idea that random sampling can be used to solve deterministic mathematical problems. They recognised that development of digital computers would enable Monte Carlo methods for many applications [11].

In its most simple form, a Monte Carlo simulation consists of a large number of random trials. Information is obtained by tabulating the results of these trials. Therefore, a Monte Carlo simulation randomly selects a point somewhere is the search space and all points are used to find out information about the search space. All random moves are accepted such that a different region of search space is selected for sampling at each step. This procedure has use in some problems, but the probability of finding an optimal solution can be very low in problems with high numbers of possible solutions (large search space) and a relatively small number of optimal solutions.

If the search space is considered analogous to an energy surface, such as a collection of helium atoms in a cube, the position of each atom is described by three parameters that give its coordinates within the cube. The energy of this system is given by the sum of all pair-wise interaction energies. When calculating the average energy of this system, a simple Monte Carlo simulation should not be used because a random placement of the atoms may, at some point of the simulation, place two of the atoms so close together

that their interaction energy is virtually infinite. The close separation adds an infinite energy to the ensemble of atom distributions and produces an infinite average energy.

In the real world, two helium atoms would never get that close together. Therefore, a modification to the simple Monte Carlo simulation can be made so that unrealistic samples are not placed into the ensemble. Metropolis et al [12] proposed a new sampling procedure which incorporates a measure known as the *temperature* of the system. The modified Monte Carlo method is known as a Metropolis Monte Carlo simulation. In contrast with the simple Monte Carlo simulation, a new point in search space is sampled by making a slight change to the current point.

In the example used here, making a random, small change to each atom's coordinates creates a new configuration of the helium atoms. If the energy of the new configuration is less than that of the old, the configuration is added to the ensemble. If the energy rises and is small enough, the new configuration is added to the ensemble. Conversely, if the energy rise is too large, the new orientation is rejected and the old orientation is again added to the ensemble.

1.5.2. Simulated Annealing

In 1983, Kirkpatrick et al [13] proposed a method of using a Metropolis Monte Carlo simulation to find the lowest energy (most stable) orientation of a system. Their method is based upon the procedure used to make the strongest possible crystalline solid. This procedure heats the solid to a high temperature so that the atoms can move relatively freely. The temperature of the material is slowly lowered so that at each temperature the atoms can move enough to begin adopting the most stable orientation. If the material is cooled slowly enough, the atoms are able to 'relax' into the most stable orientation. This slow cooling process is known as *annealing*, and their optimisation method known as Simulated Annealing.

A Simulated Annealing optimisation starts with a Metropolis Monte Carlo simulation at a high temperature. The high temperature means that a relatively large percentage of the random steps (that result in an increase in the energy) will be accepted. Initially, the algorithm behaves not unlike a random search algorithm, sampling different parts of the search space.

After a sufficient number of Monte Carlo steps, or attempts, the temperature is decreased. The Metropolis Monte Carlo simulation is then continued and the process is repeated until a final temperature is reached. Some degree of experimentation is required to find a cooling function that performs well with a particular problem.

1.5.3. Iterative Least Squares

The method of least squares was discovered by Gauss in 1795. It has since become the principal tool for reducing the influence of errors when fitting models to given observations. Today, applications of least squares arise in a great number of scientific areas, such as statistics, signal processing, and control.

Essentially the method performs a least squares fit to a required function then iteratively improves the fit by calculating the difference between the required function and the

current approximation to it. There are many variations of this method for both linear and non-linear problems.

Generally, the method of least squares on a function containing some number of unknown constants, for instance $f(x) = ax + b$ (where a and b are not yet known), finds the values of a and b that minimise the sum of the squares of the residuals (i.e. the sum of terms of the form $(y_i - f(x_i))^2$. The method attempts to find the equation for the curve, $y = f(x)$ of the required form, that best fits the data points (x_i, y_i).

The text by Bjork [14] provides a comprehensive survey of the available literature on the subject.

1.5.4. Projection Methods

Projection methods [15,16,17] attempt to obtain a 'mini-max' fit to the required far field power pattern. Using an initial distribution, these methods find the worst deviation between the far-field pattern and the requirement. The methods then implement the least aperture adjustment that will achieve the required specification at this far-field point. This process is then repeated until the pattern converges. Constraints on amplitude and phase distributions can be included.

1.5.5. Discussion of Techniques

As shown above, the solution search space size in array pattern optimisation can grow to be very large, dependant on the number of elements and the degree of digital control available for tuning the amplitude weights and phase settings. Given the size of this search space, a few thousand runs of either of the Monte Carlo methods is unlikely to find any optimal solutions. The technique can be useful for gaining statistics on the distribution of solutions available, but is unreliable as an optimisation method on large problems.

Simulated annealing (SA) is designed to function primarily as an optimiser and has been used before in the field of antenna performance optimisation. Simple minimisation algorithms find only local minima, but the mechanisms of simulated annealing methods avoid settling prematurely on local minima. The system temperature controls the convergence and so the choice of cooling function can influence the value of the final optimised solution. Rodriguez et al [18] optimised the performance of a cylindrical array radiation pattern using SA. The authors used a weighted sum cost function to combine desired features in the radiation pattern such as a desired sidelobe level and mainlobe beamwidth. They managed to produce a single pattern for an 8 x 8 element array. The pattern achieved -25dB sidelobes (with the mainlobe normalised to 0 dB). The authors claimed that the technique avoided convergence on local minima. More recently, López et al used SA to produce a double difference pattern with five inner sidelobes [19]. The final synthesised pattern achieved -20dB sidelobes between the two mainlobes.

Simulated annealing is useful optimisation technique but the cooling function dictates how well the optimiser can avoid premature convergence. If an energy level is too low, the current state cannot be escaped from and convergence on a local minima follows. The optimisation goals such as low sidelobe level and beamwidth must be combined

into a single scalar parameter, inevitably biasing the convergence (see Chapter 6). The SA provides a single solution at the end of each run which means that multiple runs are required if more than one optimal solution exists (as can be the case when there are numerous optimisation goals due to the trade-off between objectives). There are also no means of ensuring that subsequent runs will find other substantially diverse solutions unless the cost function is modified. Diversity in solution sets is important to the engineer as it may highlight acceptable and usable solutions that would otherwise be unknown.

Iterative Least Squares methods have been employed by Vaskelainen [20] who used them to produce a number of different radiation pattern shapes with a conformal array antenna. The disadvantage is that a required known pattern is required, the so-called 'destination function'. In the absence of a priori knowledge, a desired destination function could prove to be unrealistic (prior knowledge such as maximum possible achievable gain can help here). The technique can be prone to convergence on local optima. In least squares, the covariance matrix must be recalculated during each iteration which can be computationally demanding, particularly if the array size is large.

A paper by Guy et al [21] gives an overview of pattern synthesis techniques for conformal arrays. The paper demonstrates the use of the successive projection method to produce a low-sidelobe difference pattern, but the algorithm is described as being prone to convergence on local optima and therefore requiring several repeated starts to ensure good solutions.

The main advantage of the successive projection technique is that only one point is optimised at a time without the need to invert a large matrix. It is also well suited to giving a 'mini-max' fit to the requirement, without the need to adjust any synthesis parameters. The problem is that the method slows linearly with the number of elements.

The earlier cited paper using a Generalised Projection technique by Bucci [15] suggests that the technique will need generalising or that penalty function will be have to be included as an objective in order to constrain excitation set values. This point is important, as excitation sets with high dynamic ranges can promote mutual coupling between the elements in real systems. Bucci's technique was reportedly fast, but again only provides a single solution.

The Monte Carlo and SA approaches perform global searches of the solution space, but of the two, SA is preferred as its search is controlled by the temperature measure. All the above numerical optimisation techniques operate using a single optimisation goal or measure and while they are undoubtedly popular, they all increase in complexity as the array size is increased. The application of these techniques in antenna optimisation appears to becoming less frequent as more modern numerical optimisation algorithms are appearing and are providing good results to difficult problems. The next section assesses the suitability of modern computational techniques for use in antenna optimisation.

1.6. Modern Optimisation Methods

This section looks at modern techniques that fall within the area of computational intelligence (sometimes called soft computing). Computational Intelligence is a umberella term that groups together many different types of computer based methodologies and algorithms.

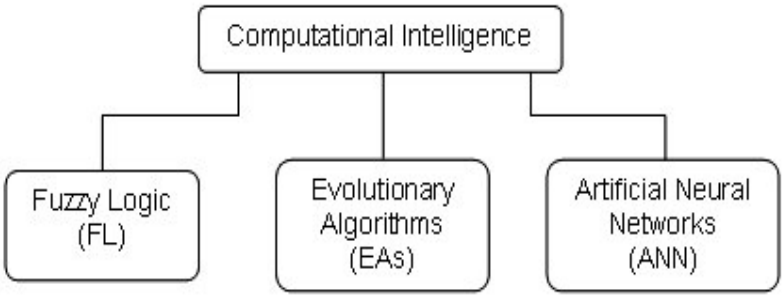


Figure 1-1 Computational Intelligence Techniques.

Figure 1-1 shows three common areas within Computational Intelligence. These three main areas are fuzzy logic, evolutionary algorithms and artificial neural networks. Fuzzy logic and Artificial Neural networks are mapping (classifier) methods, rather than optimisation tools, but they have been applied in the field of array pattern optimisation.

Fuzzy logic [22] is a method for processing uncertain or noisy input data into crisp decisions or control signals for a system. While it is a powerful technique in its own right, it has limited applicability in antenna pattern optimisation where the input parameters (such as excitation sets) are more certain. It is not uncommon for the actual design of fuzzy systems to require optimisation before good results are obtained [23,24].

An Artificial Neural Network (ANN) [25,26,27] is an information-processing paradigm that is inspired by the way biological nervous systems, such as the brain, process information. The key element of this paradigm is the novel structure of the information processing system. It is composed of a large number of highly interconnected processing elements (neurones) working in unison to solve specific problems. ANN learn by example in much the same way humans learn. ANNs are configured for a specific application, such as pattern recognition or data classification through a learning process. ANNs are very popular in the research community and their main strength lies in their ability to identify patterns or trends in data.

ANNs have been used in antenna performance optimisation although a literature search only uncovered a small number of examples. Aboul-Dahab et al [28] used an ANN in the receive chain of a linear array antenna to weight a received pattern to reduce sidelobe levels. The results were very impressive - low sidelobe patterns were produced with up to -80dB sidelobes. The authors state that the training time of the ANN rises with the array size and do not mention how the pattern degrades if the array is steered away from boresight. The technique only works on receive patterns (sidelobe cancellation through monitoring of the signal to noise ratio).

A similar technique by Reza and Chrostodulou [29] trained an ANN that took a radiation pattern as an input and as an output produced a design for a linear array and a set of weights to achieve the pattern. The array solution contained the least number of elements required to reproduce the pattern. The use of an ANN in this area is a novel technique, but assumes that only one pattern is needed. It also assumed that the

elements were isotropic so it is not clear how mutual coupling and element patterns would affect the results. The technique requires knowledge of an achievable radiation pattern.

Its clear that ANNs have application in this area and that they can be formulated to model non-linear problems but ANNs themselves are not optimisers. One disadvantage of ANNs is the fact that individual relations between the input variables and the output variables are not developed by engineering judgement so the ANN tends to be a 'black box' system or input/output table without analytical basis. As such ANNs are not good at providing understandable knowledge on how a problem is solved. Also the computation time to develop and train a neural network can be demanding, particularly on large problems.

The other major subset of computational intelligence is the area of evolutionary algorithms (EAs). EA is another umbrella term used to describe a number of computer-based problem solving systems which use computational models of some of the known mechanisms of evolution as key elements in their design and implementation. A variety of EAs are in existance. The most popular variants are genetic algorithms (GAs), Evolutionary Programming (EP), Differential Evolution (DE), Evolution Strategies (ES), Genetic Programming (GP), Population Based Incremental Learning (PBIL), Particle Swarm Optimisation and Ant Colony Optimisation (ACO).

They all share a common conceptual base of simulating the evolution of individual structures via processes of selection, reproduction and mutation. The processes depend on the perceived performance of the individual structures as defined by an environment.

More precisely, EAs maintain a population of structures, that evolve according to rules of selection and 'genetic operators'. Examples of genetic operators include 'recombination' and 'mutation'. Each individual in the population receives a measure of its fitness in the environment. Reproduction focuses attention on high fitness individuals, thus exploiting the available fitness information. Recombination and mutation perturb those individuals, providing general heuristics for exploration of the search space. The common EA variants are shown below in Figure 1-2.

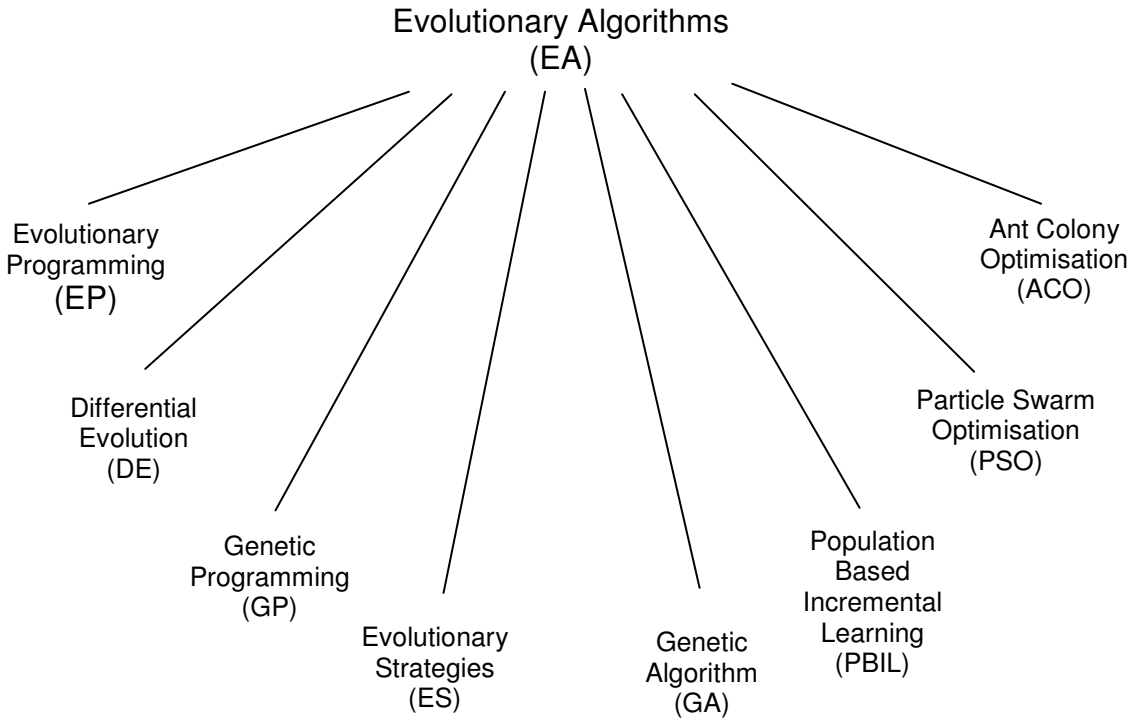


Figure 1-2 Overview of EAs

1.6.1. Genetic Algorithm

The Genetic Algorithm (GA) is a model of machine learning which derives its behavior from a metaphor of some of the mechanisms of evolution in nature [30, 31]. The algorithm creates a population of individuals represented by chromosomes. The chromosome is a string of variables that is analogous to the chromosomes present in nature. The individuals in the population then go through a process of simulated evolution modeled on the Darwinian theory of natural selection.

Genetic algorithms are used in a number of different application areas. One example is in multidimensional optimisation problems in which the chromosome can be used to encode the values for the different parameters being optimised.

In practice the genetic model of computation can be implemented by having arrays of bits or real valued numbers to represent the chromosomes. During the evolutionary model, the genetic operators of crossover and mutation modify the chromosomes. The detail of this process is given in Chapter 4.

One iteration of this algorithm is referred to as a generation. The first generation of this process operates on a population of randomly generated individuals. From there on, the genetic operations, in concert with the fitness measure that defines the measure of success of each individual, operate to improve the population.

A selection mechanism is used to choose individual members from the current generation as parent solutions for the subsequent generation. Solutions with the highest

fitness tend to be selected more often and hence pass on their genetic information to their offspring. This exchange strengthens the population over time until it converges on a solution.

1.6.2. Evolutionary Programming

The term 'Evolutionary Programming' was originally conceived by Lawrence J. Fogel in the 1960s [32] although the general idea of using a computer to simulate evolution appeared in primitive forms throughout the 1950s [33]. It is a stochastic optimisation strategy similar to genetic algorithms, but instead places emphasis on the behavioural linkage between parents and their offspring, rather than seeking to emulate specific genetic operators as observed in nature. Evolutionary programming is similar to evolutionary strategies (Section 1.6.5), although the two approaches were developed independently.

In all Evolutionary Programming methods, each member of the current population is used to generate an offspring so the selection mechanism used to choose parent solutions in the GA is not used. Each offspring is placed into a new population. When all offspring have been generated, the current population is merged with the new population of offspring, and a separate selection procedure is used to generate a current population for the next generation.

In EP and GAs, there is an underlying assumption that a fitness landscape can be characterised in terms of variables, and that there is an optimum solution (or multiple such optima) in terms of those variables. For example, if one were trying to find the shortest path in a Travelling Salesman Problem, each solution would be a path. The length of the path could be expressed as a number, which would serve as the solution's fitness. The fitness landscape for this problem could be characterised as a hyper-surface proportional to the path lengths in a space of possible paths. The goal would be to find the globally shortest path in that space, or more practically, to find very short tours very quickly.

The basic EP method involves 3 steps:

(Repeat the steps until a threshold for iteration is exceeded or an adequate solution is obtained):

(1) Choose an initial population of trial solutions at random. The number of solutions in a population is highly relevant to the speed of optimisation, but no definite answers are available as to how many solutions are appropriate.

(2) Each solution is replicated into a new population. Each of these offspring solutions are mutated according to a distribution of mutation types, ranging from minor to extreme with a continuum of mutation types between. The severity of mutation is judged on the basis of the functional change imposed on the parents.

(3) Each offspring solution is assessed by computing its fitness. Typically, a stochastic tournament (see Section 4.3) is held to determine N solutions to be retained for the population of solutions, although the retention is occasionally performed

deterministically. There is no requirement that the population size be held constant nor that only a single offspring be generated from each parent.

It should be pointed out that EP typically does not use crossover as a genetic operator. Mutation is used but it simply changes aspects of the solution according to a statistical distribution. This distribution weights minor variations in the behaviour of the offspring as highly probable and substantial variations as increasingly unlikely.

Further, the severity of mutations is often reduced as the global optimum is approached (similar to the reduction in temperature of simulated annealing).

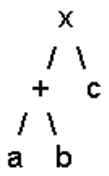
1.6.3. Differential Evolution

The main idea behind Differential Evolution (DE) [34] is a scheme for generating trial parameter vectors. The basic strategy is that the weighted difference between two randomly selected solutions from the population is used as the source of a random variation for a new trial solution. DE has been shown to outperform variants of SA on certain test cases.

1.6.4. Genetic Programming

Genetic Programming (GP) provides a method for automatically creating a working computer program from a high-level problem statement of the problem [35,36]. Genetic programming achieves this goal of automatic programming (also sometimes called program synthesis or program induction) by genetically breeding a population of computer programs using the principles of Darwinian natural selection and biologically inspired operations. The operations include reproduction, crossover, mutation, and architecture-altering operations patterned after gene duplication and gene deletion in nature. Genetic Programming can search the space of possible computer programs for an individual computer program that is highly successful in solving (or approximately solving) the problem at hand.

Genetic programming is the extension of the genetic model of learning into the space of programs. That is, the objects that constitute the population are not fixed-length character strings that encode possible solutions to the problem at hand, they are programs that, when executed, are the candidate solutions to the problem. These programs are expressed in genetic programming as parse trees, rather than as lines of code. Thus, for example, the simple program "$a + b \times c$" would be represented as:



or to be precise, as suitable data structures linked together to achieve this effect. The programs in the population are composed of elements from two sets known as the function set and the terminal set. These sets typically contain fixed symbols selected to be appropriate to the solution of problems in the domain of interest.

In GP the crossover operation is implemented by taking randomly selected subtrees in the individuals (selected according to their fitness) and exchanging them. GP usually does not use mutation operators.

1.6.5. Evolutionary Strategies

Evolution Strategies (ES) are in many ways very similar to EP and to GAs. As their name implies, ES too simulate natural evolution. The differences between EPs and ES arise primarily because the original applications for which the algorithms were developed are different. Similarly, while GAs were designed to solve discrete or integer optimisation problems, ES were applied first to continuous parameter optimisation problems associated with laboratory experiments.

ES were introduced in the 1960s by Rechenberg [37] and further developed by Schwefel [38]. The first attempts at using ES to solve discrete optimisation were also made by Schwefel [39].

In a GA, mutation is usually a background operator, fixed in value and applied with low probability, and crossover is the primary search mechanism. In ES each variable has an adaptive mutation rate that is usually normally distributed with a zero expectation. Therefore when optimising ten values, a further ten mutation rate variables are required.

This mutation mechanism enables the ES algorithm to evolve its own mutation strategy parameters appropriate to the problem being tackled as the search progresses, a process termed self-adaptation by Schwefel [40]. Like EP, considerable effort has focused on adapting mutation as the algorithm runs. Unlike EP, however, recombination does play an important role in evolution strategies, especially in adapting mutation.

A variety of recombination operators have been used in ES. Some, like the GA crossover operator, combine components from two randomly selected parents, while others allow components to be taken from any of the solutions in the parent population. Recombination is applied not only to the control variables but also the strategy parameters. Indeed, in some ES implementations different recombination operators are applied to different components of the solution representation.

1.6.6. Population Based Incremental Learning

Population Based Incremental Learning (PBIL) [41] is a guided search algorithm that obtains its directional information from the previous best solutions. Typically a binary string is used to encode the optimisation variables and a real-valued second string, known as the prototype vector is generated of the same length as the binary string. At the start of the search, the prototype vector values are all set to 0.5.

As the search progresses, each element of the probability vector is updated by small increments so as to favour the generation of either a one or a zero for the corresponding bit in a trial solution vector. (The initial value of 0.5 provides zero bias). To avoid the chances of being trapped in local minima the probability vector is allowed to mutate by a small amount at each generation of the algorithm. A number of empirically derived tuning parameters are required in order to promote good performance. For example, the

number of trial vectors that are generated and evaluated before updating the probability vector must be chosen (typically 100).

A term known as the learning rate determines the probability vector update increment (typically 0.1). Smaller values result in wider searches, but slower convergence. Similarly a negative learning rate can be used to distance the probability vector from the worst performing solutions (typically 0.075). The probability and amount of mutation must also be set for the probability vector. It is possible to implement a PBIL algorithm using only twenty lines of code (excluding the objective function evaluation) [42].

1.6.7. Particle Swarm Optimisation

PSO algorithm was introduced by Russel Eberhart (an Electrical Engineer) and James Kennedy (a Social Psychologist) in 1995 [43]. PSO belongs to the categories of Swarm Intelligence techniques and Evolutionary Algorithms for optimisation. It was inspired by the social behaviour of birds, which was studied by Craig Reynolds (a biologist) in late 80s and early 90s. He derived a formula for representation of the flocking behaviour of birds. The formula was later used in computer simulations of virtual birds, known as Boids. Ebenhart and Kennedy recognised the suitability of this technique for optimisation and came up with the Particle Swarm Optimiser.

The representation of the optimisation problem is similar to the encoding methods used in GAs. The main difference is in the search mechanism. In PSO, the variables are called dimensions that create a multi-dimensional hyperspace. "Particles" fly in this hyperspace and try to find the global minima/maxima, their movement being governed by a simple mathematical equation. PSO has no evolution operators such as crossover and mutation. Each particle has a position and velocity associated with it and hence requires the same storage as ES.

PSO has been successfully applied in many areas: function optimisation [44], artificial neural network training [45] and fuzzy system design [46].

1.6.8. Ant Algorithms

Ant algorithms (or Ant Systems) are a novel technique first developed by Dorigo et al in 1991 [47]. The Ant System is a population-based approach. In this respect it is similar to genetic algorithms although there is not a population of solutions being maintained. Rather, there is a population of ants, with each ant finding a solution and then communicating with the other ants in the hope it will help them find even better solutions.

When ants move, they leave a trail of pheromone in their wake. This pheromone is detected by other ants which in turn follow the trail. If one imagines an ant’s nest and a source of food some distance from the nest, the ant has many routes it can follow to get from the nest to the food source. If a single ant takes a long route to the food source, its pheromone trail will be weaker than if it were to take a short route, as pheromone intensity reduces with time (evaporates). If more ants are introduced, over time, the shorter route will be followed more often than the weaker route as the pheromone builds up on it. Over time, the route will become the major path to the food. This ant colony behaviour was modelled in software and the majority of research has been on

Travelling-Salesman type (route distance minimisation) problems [48]. Some of the finer details of the algorithm are left out here for brevity, but there are many sources of literature available on the internet at the time of writing [49,50].

1.7. EA Literature Review

In recent years, evolutionary algorithms have become very popular in electromagnetic optimisation problems.

A study of antenna array optimisation literature has shown genetic algorithms to be the most popular of the EAs although there are a small number of examples of other EA types being applied. Table 1-1 shows the number of unique examples found during a search of the INSPEC and Compendex databases.

Total Papers	2	7	0	6	0	1	5	>150
EA	Ants	EP	ES	DE	GP	PBIL	PSO	GA

Table 1-1 Application of EAs in Antenna Array Performance Optimisation

A paper by Fogel [51] suggests that the reasons for the popularity of evolutionary computation techniques include the flexibility of the procedures, as well as the ability to self-adapt the search for optimum solutions on the fly. Fogel predicts that as desktop computers increase in speed, the application of evolutionary algorithms will become routine.

Ant Algorithms have been used by Karaboga et al to perform null steering in linear antenna arrays [52]. The authors use ant colony optimisation (ACO) to vary element positions in the array in order to introduce nulls in the radiation pattern. The technique shows promise but there has been no further research uncovered. Varying element positions is not practical in real arrays as they are fixed during manufacture, and one of the advantages of active array technology is the fact that there are no mechanically moving parts required to produce or steer the beam. The majority of research into ACO has been on TSP and route planning optimisation problems and they do not immediately lend themselves to application in other areas.

Evolutionary Programming is easier to apply to array pattern optimisation problems as the chromosome can directly contain the excitation set. There were still relatively few examples of their application in this area (although seven papers were found, they were all by the same collections of authors). The most recent was a paper by Hoorfar [53], who used a hybrid EP-GA technique to optimise the gain of a six-element yagi array. The method converges to provide a single solution and worked by varying the distances between elements which is only practical if one radiation pattern is required.

No examples of Evolutionary Strategies in array control optimisation were found - probably attributed to the ease of applicability of other EA techniques and their success in electromagnetic optimisation problems.

Differential Evolution has been applied to linear array antenna optimisation problems, most recently in September 2003 by Kurup et al [54]. Kurup varied element positions

and compared the results to the variation of element phase values in order to reduce sidelobe levels in the pattern. A cost function was used that combined the sidelobe measure and the value of mainlobe gain. The work concludes that joint optimisation of the element positions and phases are better than phase only optimisation. This technique would limit the final array to a smaller range of patterns due to the optimised element positions. The algorithm converges to provide a single solution.

Genetic Programming is intended to evolve computer programs, and while this EA has many applications in other domains it is not immediately obvious how it can be applied in array performance optimisation, hence the lack of research papers found. One could imagine use of GP to evolve a function for describing excitation set values.

Population Based Incremental Learning was used by Horrell [55] in a procedure that attempted to produce radiation patterns within a certain boundary mask. PBIL was used to optimise a cosec^2 antenna pattern produced by a linear array antenna. The main problem with this application of PBIL is that knowledge of achievable radiation patterns were required in order to define the mask boundaries and without this knowledge, several runs are required. The objective function minimised the distance between the achieved pattern and the mask.

Particle Swarm Optimisation has been used by Boeringer and Werner who compared the performance of PSO with that of a GA on a linear array problem [56]. A mask was again used to enable a simple objective function calculation and it was shown that the GA outperformed the PSO. In a similar paper, Giles and Rahmat-Samii demonstrated PSO to be capable of simultaneously optimising two patterns required from a single twenty element linear array [57]. Good solutions for both patterns were found after around 750 iterations. The cost function was based upon the difference between the achieved pattern and a mask. The authors commented that the performance is likely to reduce as the array size increases and the problem becomes more difficult to solve.

The application of Genetic Algorithms has received far more attention in array performance optimisation, and it has been practised since the early 1990's. By comparison PSO is a relatively new technique (1995).

The earliest example of the application of genetic algorithms to antenna array optimisation found was by Haupt et al [58] in 1993. Haupt used a GA to thin a 50 element linear array of isotropic elements. A single fitness measure was used equal to the highest calculated sidelobe level. The algorithm discarded the weakest 50% of the population at each generation. This practice is rarely used today as the probability of crossover helps to maintain good performing solutions (it is still commonly used in EP and ES). The GA provided a single solution to the problem (i.e. converged to a single point on the cost function). As the algorithm maintained the best solutions found, it was enforcing an 'elitism' operator which ensures good solutions cannot be lost once found. This work stood as an important first step and over the next decade, many researchers followed and advanced Haupt’s work by optimising other types of arrays and different operating variables.

The chromosomes used in Haupt's paper simply switched elements on or off. Later work by both Haupt and many other researchers optimised amplitude weightings, phase control and some aspects of array geometry:

O'Neill [59] optimised element placement in thinned arrays using genetic algorithms (in much the same way as Haupt) the main difference was the use of a dedicated elitism operator to ensure the best solution found in each generation was not lost (rather than discarding half the population at each generation).

Shimizu [60] used GAs to optimise the performance of an eight element linear array with each element controlled by a four bit phase shifter. Shimizu used a uniform amplitude distribution and optimised the phase values. The work demonstrated that GA optimisation works for steered arrays (steered 30° off boresight in this case). While the technique worked, it was only capable of optimising the main beam position. The crossover scheme used was unusual as it randomised the two least significant binary digits of ninety of the one hundred chromosomes in the population. Eight of the remaining ten individuals were formed by applying this modified crossover operator to the elite solution and the final two were formed by completely randomising two more individuals. In effect, this algorithm is an entirely mutation based algorithm coupled with elitism, with no recombination or selection at all. The algorithm is more like EP than GA.

In 1997 Johnson and Rahmat-Samii [61] summarised the use of GAs in electromagnetic optimisation problems and proposed the use of them on large planar array antennas. The paper is recommended as a basic introduction to GA techniques in electromagnetics. Haupt furthered his earlier work by comparing GAs to gradient search methods and concluded that gradient search methods only perform well on small arrays and that GA methods perform substantially better than gradient methods on larger arrays [62]. He later used GAs to optimise amplitude tapers that are applied to set subarrays within a linear array [63].

All the early GA work cited above concentrated on forming and optimising a single mainlobe, formed using a linear array. Marcano et al [64] optimised multiple beam linear antenna arrays using GAs. A GA was used to determine the weights that would result in a radiation pattern with two mainlobes and low sidelobes. The arrays were small containing just 20 elements so the search space was not particularly large. Later the same year, the authors furthered their work by optimising patterns with three mainlobes [65].

Alphones and Passoupathi [66] looked at element positions rather than weights. They used a GA to vary the spacing between elements on a linear array antenna. The objective was to produce nulls in the radiation pattern at desired positions. Again the GA performed well as the array size and search-space were reasonably small.

Ares et al [67] demonstrated application of genetic algorithms and simulated annealing techniques in optimising the aperture distributions of antenna arrays. Their work demonstrated GAs to be faster than simulated annealing in linear array optimisation.

All the work so far cited relied on one or two measures of fitness to guide the GA to convergence. In most cases a single objective measure was used but occasionally weighted sums of objectives were used. Some authors stated that the fine-tuning of the weights or particular method for combining objective values strongly influenced the success of the optimisation exercise.

A number of new GA based algorithms have been published in more recent years that remove the need to combine objectives into a single cost function. These algorithms are known as multiobjective genetic algorithms (MOGAs). The main difference between MOGAs and GAs is that MOGAs converge to give a trade-off surface of possible solutions (more correctly known as a Pareto-optimal set) rather than a single solution as is the case with the simple GA.

A rare example of true consideration of multiobjective optimisation of antenna performance was proposed by Weile et al in 1996 [68] in "Multiobjective synthesis of electromagnetic devices using nondominated sorting genetic algorithms". The work suggested that a microwave absorber design could be optimised using the Non-dominated Sorting Genetic Algorithm (NSGA) developed by Srinivas and Deb [69]. Weile goes on to demonstrate NSGA on a microwave absorber optimisation task. The work suggests, but does not demonstrate an application to antenna arrays. Some NSGA solutions are reportedly lost during the run and the results obtained were not necessarily as diverse as they could be.

Weile and Michielssen [70] extended the application of multiobjective algorithms to antenna arrays in 1996 when they used NSGA to trade-off beamwidth and sidelobe level on a linear array antenna of 200 isotropic elements. They also tested the technique on a planar array of 16 x 16 elements. The algorithm was run for 1000 generations and delivered a curve of beamwidth against sidelobe level, with beamwidth varying approximately proportional to sidelobe level. Patterns were found by using the GA to either thin the array or alter phase values.

Surprisingly, since the work of Weile no other examples have been found in the literature that progresses the application of multiobjective algorithms in antenna array optimisation.

More recent work (1998 onwards) has advanced the application of the GA, mainly to linear array optimisation problems. Brann and Virga [71] generated optimal excitation sets for single-ring cylindrical arc arrays. Both amplitude and phase values were found for the simple arc arrays. A simple GA was used that was guided by a weighted sum of directivity and null depth, with the directivity measure weighted much higher than the null depth measure. Using weighted sums of objectives can prevent regions of the search space from being explored and is not recommended for detailed trade-off studies (see Chapter 6).

Lozano et al [72] developed a genetic algorithm procedure for linear array failure correction. The procedure generated amplitude weights for the array after some simulated element failures were included. The work is similar to earlier work by Haupt

(cited above) and is analogous to thinning an array and then calculating appropriate weights to achieve low sidelobe performance and maximisation of antenna gain.

Lopez et al [73] reduced sidelobe levels in linear arrays by varying the phase of a small number of elements in a uniformly excited array. With the mainlobe normalised to 0 dB, maximum sidelobe levels reduced from a peak of –13dB to around –20dB by varying the phases of just 20 elements in an array of 100. Sidelobe values of –25dB were achieved by modifying 44 excitations. Varying a smaller number of excitations reduces the search space and so increases the GA's chances of finding good solutions. It may however, reduce the chances of finding the optimal solution. It would be a good idea for further work to optimise the number and location of elements chosen for phase optimisation and the actual phase shifts to be applied to them.

Bregon and Fernandez del Rio [74] presented a software package in 2002 that combines a simple GA with a gradient searching algorithm to improve the accuracy of final answers. A case study is presented on a thinned 200-element linear array antenna with the objective of reducing the maximum sidelobe level. The paper does not state the difference between the best solution achieved using the GA, but instead quotes the final answer achieved after the gradient search method had operated. Again with the mainlobe normalised to 0 dB, sidelobe levels of –23.92dB were achieved with 73% of elements operating.

There is much less published on the subdivision or partitioning of array antennas. The common approach for the array partitioning is to randomise a number of similarly shaped groupings of elements [75]. There is no evidence in the wider literature that randomising groupings is in any way optimal, although the trial and error method demonstrated that a good compromise between sum and difference patterns is achievable using a subdivided array. A more in depth investigation of subarray literature is given in Chapter 8.

Lopez furthered his work in 2001 by using a simple GA to optimise subarray weighting for the difference patterns of monopulse antennas [76]. The paper is similar to the early work of Haupt except that the actual subarray configurations are optimised simultaneously with the amplitude weights. The technique is demonstrated on a linear array antenna and while extension to planar arrays is suggested, no demonstration is given. The technique as it stands could not be readily applied to planar arrays as the method suggested for dividing the linear array into subarrays would not work in two dimensions.

The GA has been shown to give good performance in antenna array optimisation. The manner in which the GA has been applied to pattern synthesis problems has been consistent - the results studied suggest more experimental work is needed to improve its performance. The area of multiobjective optimisation appears to show great promise and is worthy of further research.

From the review of prior work it seems that there are a number of areas that require further research:

- The optimisation of the control of large antenna arrays appears to be a limiting factor with EA techniques
- The method of representing the antenna array problem and formulation of cost function seems to have shown little advancement since the first applications of GAs in the 1990's.
- The optimisation of conformal array performance has not been thoroughly investigated using modern optimisation techniques.
- Multiobjective optimisation of array pattern performance has not been researched in any detail. All prior work with the exception of Weile [66] has used a combined cost function and provided no Pareto trade-off surfaces. Weile's work was limited in scope as it addressed only two performance objective measures, and used an early generation MOGA. The earlier cited Horrel stated in [53] "In a nutshell…..(the optimiser) finds what you ask it to, not necessarily what you want". It is a hypothesis of this thesis that by using a multiple objective algorithm, it is possible to be more concise about 'what you want', promoting a more efficient search.
- Optimisation of subarray design and impacts on array pattern performance have not been subject to modern optimisation techniques.

1.8. Statement of Aims

From the review of prior work and the identification of potential research areas, the aims of this thesis are to:

1. Find a method of applying EA techniques to the optimisation of large planar and conformal (two and three-dimensional) antenna arrays, that achieves good performance, irrespective of the array size.

2. Progress the area of conformal array antenna performance optimisation.

3. Further the application of evolutionary multiobjective performance optimisation of antenna array performance.

4. Find a method for the optimal subarray partitioning of antenna arrays irrespective of their geometry.

1.9. Thesis Outline & Novel Work Undertaken

This first chapter has provided a brief background to this work and states the main aims of the research. This section summarises the rest of the thesis content and highlights the main achievements.

Chapter two investigates and discusses antenna performance modelling - a prerequisite for the work that follows. Some additional theory and adopted conventions related to this chapter are provided in Appendix A.

Chapter three explains the antenna software models developed for this work and their validation.

The fourth chapter explains the theory of evolutionary algorithms and Chapter five presents the results of SGAs applied to some simple linear arrays.

Chapter six describes multiobjective optimisation, and introduces multiobjective evolutionary algorithms. An application case study is presented.

Chapter seven looks at the general area of array partitioning and reviews prior work.

Chapter eight looks new methods of array partitioning and develops a new and novel technique based on the application of artificial embryology for partitioning array antennas. This work was the subject of two patent applications in 2003 [77][78].

The partitioning and array excitation is then optimised using multiobjective evolutionary algorithms. This work was published by the IEE in November 2001 [79] (also included in Appendix B).

Chapter nine looks at the optimisation of conformal antenna performance and shows that the new partitioning scheme developed in section eight is applicable to conformal arrays and provides good results. This work was published and presented at the IEE's 12th International Conference on Antennas and Propagation in April 2003 [80] (also included in Appendix B).

Chapter ten contains a number of new and novel examples of the techniques developed to improve the performance of installed antennas. The examples include 'tactical sidelobe suppression' and an example to show how conformal array performance can be optimised in a modern aircraft installation.

Finally the research is concluded in Chapter eleven and the claims for main contributions to knowledge in this problem domain are stated.

Recommendations for further research are made in Chapter twelve.

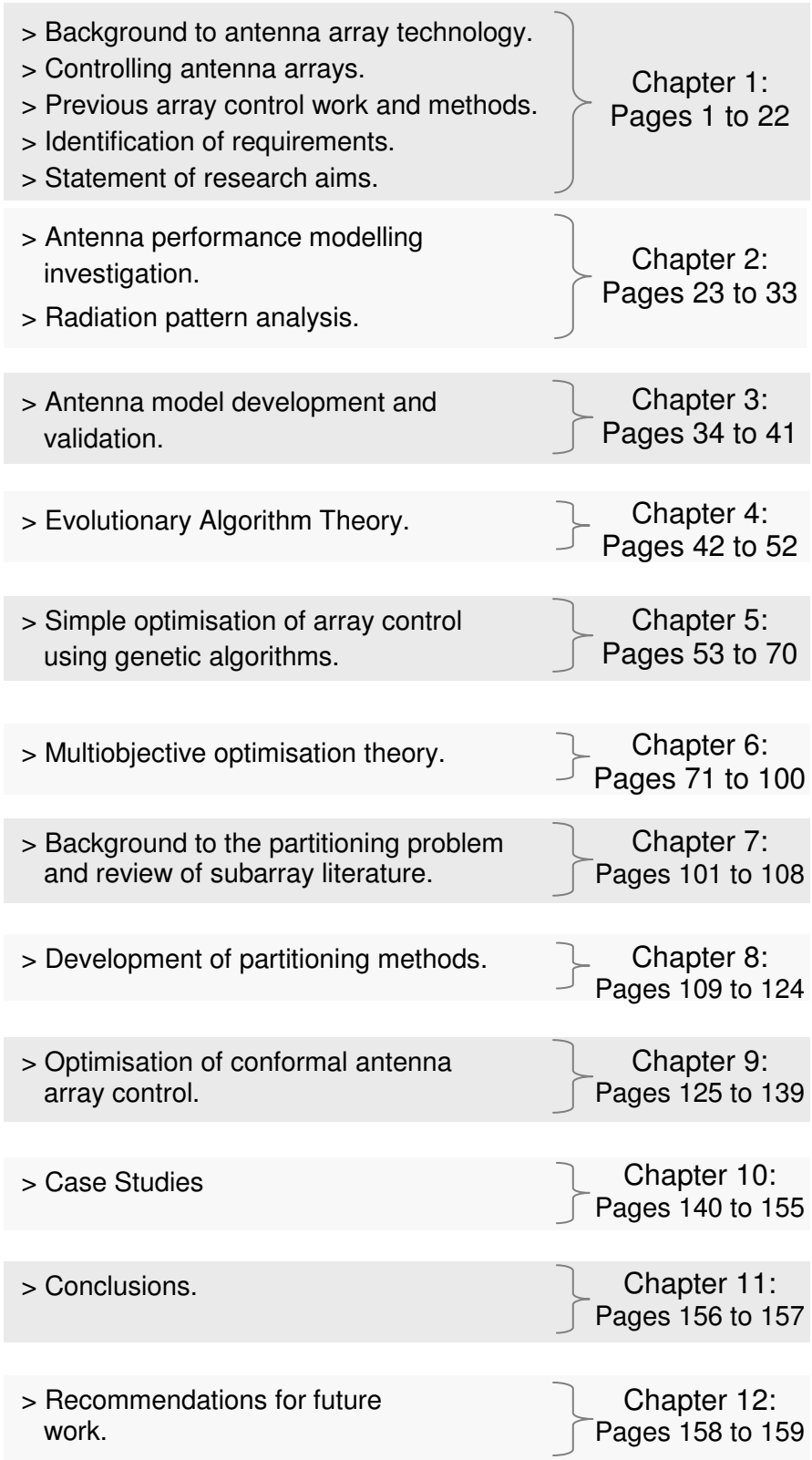


Figure 1-3 Thesis Map

CHAPTER 2
Antenna Modelling

2. Antenna Modelling

2.1. Introduction

In order to optimise antenna design or performance parameters, a representative software model is required. The model must be able to predict the radiation pattern performance of array antennas, both planar and conformal, and will form a critical stage in an optimisation algorithm by determining the success of a design solution. The level of fidelity and accuracy of the model is important if the results are to be relied upon for development of specifications for real antenna hardware.

There are many different electromagnetic modelling techniques available, the choice of which technique to use, itself presents an optimisation problem. Generally, we wish to maximise flexibility, fidelity, accuracy and precision of the antenna model but at the same instant, minimise computational burden and run times. Run time is especially important if the model is to be used with an iterative optimisation algorithm where the model may be called many hundreds or thousands of times before a good solution is found.

The Method of Moments (MoM) is an example of a theoretically exact method. The technique is well documented in the literature with several author's providing classical textbooks [81, 82, 83].

Using MoM, one can accurately evaluate the full radiation properties of any antenna if it can be subdivided into a wire or patch grid ‘segments’. The current in each individual segment is solved and the E-field and H-field are determined, by summing the overall effects from all the currents in the segments of wire and metal plate. MoM can also be applied to antennas with dielectric layers.

The equivalent wire or patch grid is assumed to sit over a free space or infinite dielectric half-plane environment. The latter choice is interesting since preliminary effects due to the ground are automatically taken into account. The main drawback of this method is the large computational resources required. As a result the MoM cannot be practically used for antennas larger than a few wavelengths. The computation time required by this technique rules out its use in iterative optimisation algorithms.

For structures with dimensions in excess of 3 to 5 wavelengths, high frequency methods such as diffraction techniques may be employed. Diffraction techniques are based on ray tracing and runtimes are less dependent on the size of the structure in wavelengths. In addition, changes in frequency do not alter run times.

High frequency methods can be subdivided into those based on Kirchoff's approximation, on corrected Kirchoff's methods and on ray tracing methods.

Some examples based on Kirchoff's methods are the Scalar Diffraction/Projected Aperture method [84], Physical Optics described in [85] and Gaussian Beam Mode Optics described in [86]. Other techniques remaining include the well-known high frequency ray tracing methods such as Geometrical Optics and the Geometrical/Uniform Theory of Diffraction [87, 88].

Upon reviewing the suitability of these methods for the modelling of array antennas, diffraction techniques were by far the most popular in optimisation literature. The diffraction techniques are less demanding of computational time, accurate enough for the purposes of this research and can be used to describe the full radiation pattern.

These high frequency methods are well suited for the analysis of electrically large array antennas as well as for the modelling of secondary effects such as the interaction of the main antenna with adjacent structures such as buildings, terrain obstacles or the host platform.

The advantages of geometrical optics and diffraction methods are fast computation times and that they can be readily applied to arbitrarily shaped surfaces with arbitrary contours, provided the surfaces and contours have radii of curvature that are large in terms of wavelength.

2.2. Array Factor

The chosen method for the antenna modelling development is based on the geometrical optics methods well covered in texts such as Skolnik's 'Radar Handbook' [89].

The modelling software developed for this work calculates radiation patterns using the superposition of multiple sources.

The array factor of an arbitrary linear array antenna of isotropic elements is given by:

$$F(\theta) = \sum_{n=1}^{N} a_n \exp(j\alpha_n) \exp[jk(x_n \sin\theta)] FE_n(\theta)$$

Eq. 2-1

where θ is the sample angle, N the total number of elements, a_n is the amplitude of the n^{th} element, α_n is the phase of element n, and x_n is the Cartesian x coordinate of element n. For real installed performance it is necessary to include the active element patterns $FE_n(\theta)$ i.e. the unique installed gain pattern of each individual radiating element (with isotropic elements all $FE_n(\theta)$ terms are identical). The element pattern should ideally be

measured once the array has been assembled and installed in its host environment so that the affects of coupling within the array and to the host platform/nearby structures are included.

Additionally,

$$k = \frac{2\pi}{\lambda} \qquad \text{Eq. 2-2}$$

and the phase values required to steer the mainlobe to θ_o is given by:

$$\alpha_n = -kx_n \sin\theta_0 \qquad \text{Eq. 2-3}$$

Eq. 2-1 is an important result and defines the factors that influence the radiation pattern and hence the variables that can be optimised for good performance. Further analysis of Eq. 2-1 shows three main components open to optimisation techniques, namely the excitation, geometry and the installed active element gain (Figure 2-1).

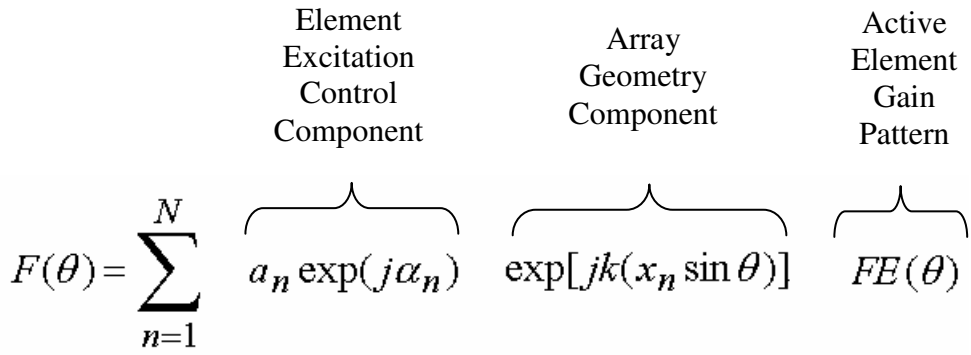


Figure 2-1 Array Factor Analysis

Similarly, these three components can be identified in the array factors of both planar and conformal array antennas.

The array factor of an arbitrary two-dimensional array is given by:

$$F(\theta,\phi) = \sum_{n=1}^{N} a_n \exp(j\alpha_n) \exp[jk(x_n \sin\theta\cos\phi + y_n \sin\theta\sin\phi)].FE(\theta,\phi) \qquad \text{Eq. 2-4}$$

with the phase to steer the beam to θ_o, ϕ_o given by:

$$\alpha_n = -k(x_n \sin\theta_o \cos\phi_o + y_n \sin\theta_o \sin\phi_o) \qquad \text{Eq. 2-5}$$

Closer examination of Eq. 2-4 shows that the additional dimension expands the geometry component, but all other factors remain the same. A similar result can be observed with the expansion of Eq. 2-4 for the conformal case:

$$F(\theta,\phi)=\sum_{n=1}^{N} a_n \exp(j\alpha_n)\exp[jk(x_n \sin\theta\cos\phi + y_n \sin\theta\sin\phi + z_n \cos\theta)].FE_n(\theta_n,\phi_n) \quad \text{Eq. 2-6}$$

The required phase taper to steer the beam to θ_o,ϕ_o is now given by:

$$\alpha_n = -k(x_n \sin\theta_o \cos\phi_o + y_n \sin\theta_o \sin\phi_o + z_n \cos\theta_o) \quad \text{Eq. 2-7}$$

Note that the active element gain pattern component $FE_n(\theta_n,\phi_n)$ must be calculated for each element in the conformal array as they all 'face' in different directions.

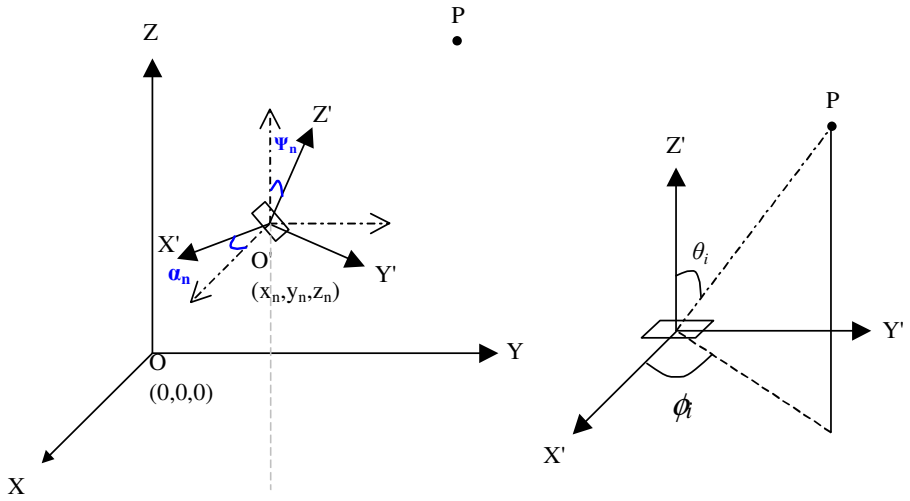


Figure 2-2 Conformal Array Modelling Geometry

Equations 2-1 to 2-7 form the far field pattern at some point *P* in the far field (identified in Figure 2-2). In order to use the correct value of individual element gain at each sample point the angles θ_i and ϕ_i must be calculated relative the element normal.

It is important to note that the equations above are only valid if the effects of mutual coupling are included. In particular the effect of the edge antenna elements is different from that of the centre elements. (Without such consideration of mutual coupling, the equations assume that the array is infinite).

While linear and planar array geometry can be described quite simply using the Cartesian coordinates of the element phase centres, conformal arrays require additional information to describe each element's rotation about the *z* axis and the direction of its normal.

2.2.1. Use of Fast Fourier Transforms

The Fourier transform is ubiquitous in engineering as it finds application in a great variety of seemingly unrelated topics.

An electrical engineering application of the Fourier transform is that the field pattern of an antenna is given by the Fourier transform of the antenna current illumination.

The fast Fourier transform (FFT) is a discrete Fourier transform algorithm. The far field pattern is calculated by applying the FFT on the corresponding element excitations. The main problem with FFTs is that the matrix to be transformed must be padded with zeros if the array geometry is irregular. The padding with zeros can increase the matrix size significantly and have a significant impact on computational efficiency, particularly when synthesising large arrays. It is also difficult to apply to conformal arrays and better suited to linear or planar configurations. For this reason the software model developed for use in this research is based on evaluation of the integrals in Eq's 2-1 to 2-6 above and does not use FFTs.

2.3. Software Implementation

Two models were developed for use in this work; the first was a linear array model based on Eq. 2-1 and the second, based on Eq. 2-6 is capable of modelling planar and conformal array configurations.

There is a trade-off between the processing time and resolution of the synthesised radiation pattern. Linear arrays of up to one hundred elements can be modelled at high resolutions (typically 0.1˚) without significant affect on run times. An azimuth cut of the radiation pattern over a range of ±90° with 0.1° resolution requires some 1810 samples.

For larger arrays some compromise is needed, particularly when working with several hundred elements or more. Generally, for analysis purposes, 0.5º resolution is acceptable as it offers a good trade-off against processing time.

MATLAB was used extensively for model and algorithm development in this work. Unfortunately, it is slow at dealing with repeated 'for-loops' forming processing bottlenecks each time such a loop is run. These bottlenecks slow the code considerably. When modelling larger planar arrays, the problem was mitigated by re-coding the processor intensive functions in C and using the new C files to generate MATLAB '.mex' files. The '.mex' files can be treated as normal MATLAB functions, but with the added benefit of a massive speed increase in this application (up to 120 times were experienced).

2.4. Analysis Methods & Performance Metrics

This section discusses metrics for the analysis of radiation pattern performance. There are many different methods of analysing the performance of a radiation pattern and they can give quite different results.

Once a radiation pattern has been calculated, the sampled data needs to be analysed in order to measure the characteristics of the pattern. The positions and magnitudes of the mainlobes, sidelobes and nulls have to be derived from the sampled data, as does the beamwidth of the mainlobe.

Antenna performance metrics are dominated by measures of gain, directivity, beamwidth and sidelobe levels. These terms are defined in Appendix B. As the measurements are ratios, the decibel scale is used throughout.

When a single value of gain is used to describe a radiation pattern, it usually relates to the gain of the mainlobe, with the term mainlobe being assigned to the lobe(s) of maximum directivity. Analysis of the mainlobe gain can be determined simply by finding the maximum of a matrix that contains the radiation pattern sample values.

Radiation patterns are often normalised to the mainlobe gain so that radiation pattern plots have their maxima at 0dB. While normalising can be useful for graphically determining relative sidelobe levels, the normalised gain levels are often omitted which can be confusing, particularly when comparing two different patterns. There are many examples of this practice in the literature [90,91,92,93,94].

Beamwidth (more specifically 3dB beamwidth) is a measurement of the angular width of the mainlobe at the point where its power level is half that of its maximum (the so-called 3dB point). Two values are needed (an azimuth and an elevation measurement) when three-dimensional patterns are being analysed. Unfortunately, calculation of the 3dB beamwidth of a mainlobe can be more difficult when using sampled radiation pattern data, as the samples rarely fall exactly on the 3dB points. Some form of interpolation is required to estimate the 3dB points. An alternative practice is to define beamwidth as the angular measurement between the nulls either side of the mainlobe. Minimisation of this separation between these nulls will reduce the 3dB beamwidth, but the magnitude of the lobe will determine the extent to which the 3dB beamwidth is reduced. Where beamwidth measures have been used in this thesis, the method used has been explained in the text

Sidelobe level analysis is more difficult to automate. For a human analyst it is easy to determine the difference between a true sidelobe and ripple occurring on the top of a mainlobe. For a computer search algorithm that it looking for maxima, additional rules are required to accurately classify them as ripples, and not as sidelobes. Throughout this work, a lobe is classed as a sidelobe if its magnitude is at least 2dB less than the mainlobe gain, and its directivity is focused away from the desired steering angle. It should be noted that this definition is not a formal one, and other authors may define sidelobes differently.

The convention for quoting sidelobe levels adopted in this work is to quote them relative to the mainlobe gain level. Sidelobes described 'as 10dB down' or '-10dB' are examples of this practice.

As the antenna model must sample a fixed angular region, it is common for part of a lobe to be described at the edges of the sampling region. Such occurrences must be treated differently in order to ensure that they are accounted for. When optimising radiation patterns, maxima at the boundaries of the sampling region have been treated as though they are sidelobe levels (see Figure 2-3).

The ‘first sidelobe’ is a term used to describe the lobe that is located nearest to the mainlobe. It is not uncommon for there to be two 'first lobes', equidistant from the mainlobe. The first sidelobe level is important in pattern analysis as it is often the maximum sidelobe level in the pattern.

Metrics that describe average sidelobe levels are less useful as they can be misleading when optimising performance. For example, when considering average sidelobe values the presence of a very high magnitude sidelobe in the pattern can be hidden by a very low magnitude sidelobe and so is not a good metric to use to guide an optimisation algorithm.

Measurement of maximum sidelobe level relies upon accurate identification of the lobes in the pattern. It is much more useful than average sidelobe level as effectively places a 'mask' on the radiation pattern that contains all sidelobes. From an optimisation point of view, measurement of a maximum is a useful objective as it is quite constraining.

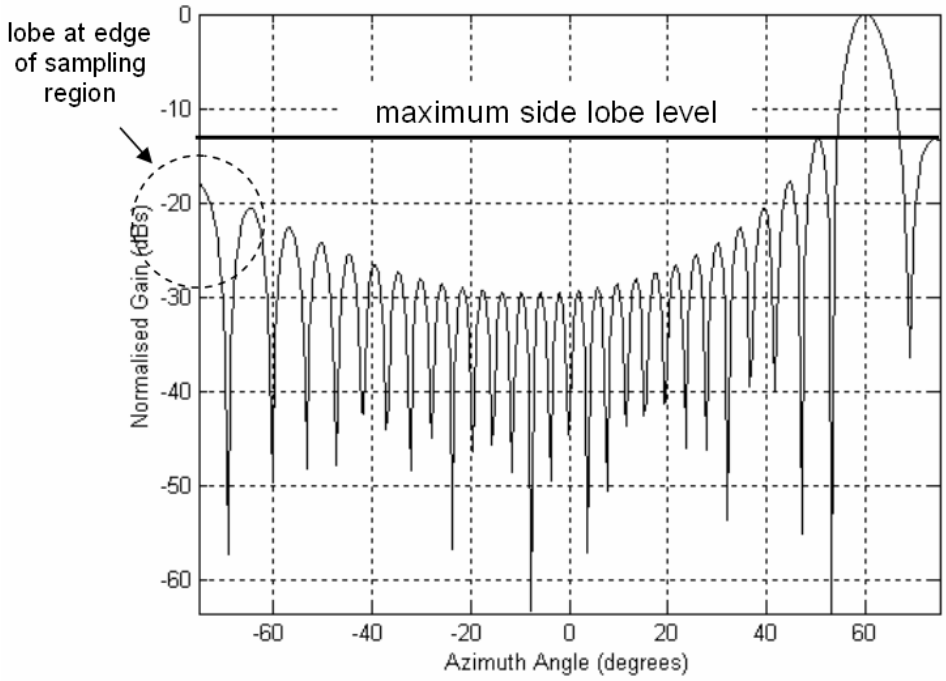


Figure 2-3 Pattern Analysis Example 1

It is common to see the use of a 'mask' in array pattern optimisation literature. There are many examples in the earlier cited literature particularly [50] and [93]. Masks must be used with care as they are not as constraining as they might first appear. Masks are often implemented in software by considering all radiation pattern samples between certain angles. For example if the entire pattern is calculated at 500 sample points, a portion of the mask can be defined by sample points 100 to 200. If a portion of the mask is not expected to contain any mainlobe sample points, there are examples in the literature of authors calculating maximum sidelobe level as the maximum sample value in the mask region.

This practice can be erroneous as illustrated in Figure 2-4 that shows a desired mask and a radiation pattern:

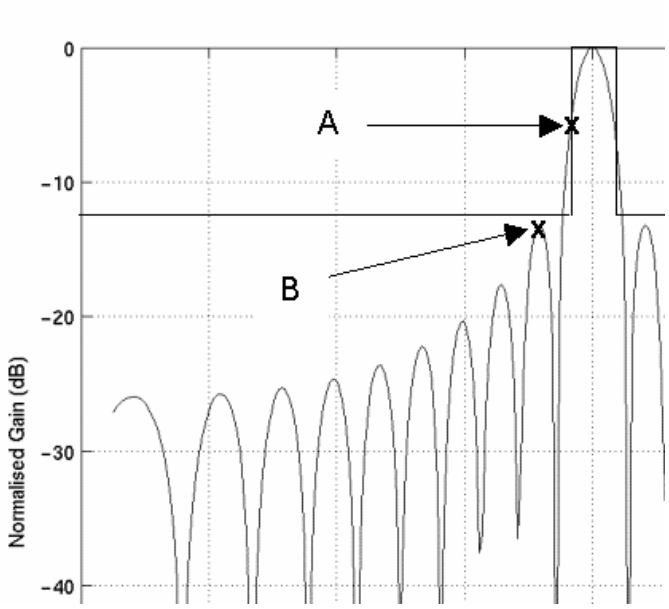


Figure 2-4 Radiation Pattern Mask

In Figure 2-4 the mainlobe has crossed a portion of the mask that is expected to be used for maximum sidelobe measurement. The sample point is incorrectly measured as the maximum sidelobe level in the pattern (the correct measurement is in fact at sample point B).

The mask around the expected mainlobe position can result in ripple appearing on the mainlobe or the so-called shoulder lobes forming (where the mainlobe and first sidelobe partially merge).

To counter this problem, it is better practice to use a dedicated algorithm to correctly identify all the sidelobes in the pattern. This approach has been adopted throughout this work, and while it is more computationally intensive, it prevents the optimisation algorithms being misguided by incorrect measurements. The algorithm for identifying the sidelobes is described below, and it increases the computational load by approximately 5% compared with the use of a simple mask.

In some patterns, particularly those produced by conformal arrays, it is useful to confine the measurement of maximum sidelobe levels to several different regions. Conformal arrays can have higher magnitude sidelobes located some distance from the mainlobe that are often higher in magnitude than the first sidelobe levels. In this work, two distinct regions have been defined for measurement of maximum sidelobe level:

- the 'near-in' region that consists of all lobes located in both azimuth and elevation within a ± 40° region around the mainlobe.
- the ‘far-out’ region that consists of all other lobes in the pattern (excluding the mainlobes).

Several software functions are required to perform the search and identification tasks and two sets of code have been written for this purpose. One set is used for two-dimensional data (radiation pattern cuts) and the second set for three-dimensional data (full pattern analysis).

When analysing three-dimensional patterns, image processing techniques were used to identify maxima and minima. For example, a two-dimensional plan view of the pattern is created and the resultant 'image' analysed using functions to locate pixels forming a maxima (corresponding to a sidelobe), and minima (a null). The functions used were ‘imregionalmax’ and ‘imregionalmin’, part of the MATLAB Image Processing Toolbox. An example of the ‘imregionalmax’ function is shown in Figure 2-4).

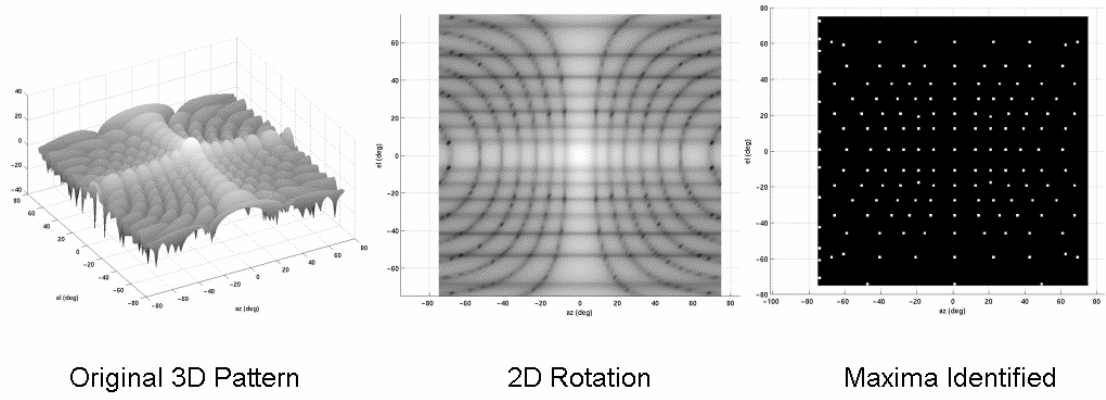


Figure 2-5 Three-Dimensional Pattern Analysis

The two functions were applied using their default settings which check eight neighbouring points around any sample point to classify minima and maxima.
The evaluation of a radiation pattern adds to the computational complexity of the modelling method so the use of efficient image processing algorithms helps minimise the total run time.

2.5. Development of Unique Installed Element Gain Patterns

The installed (or embedded) element gain patterns are present in all of the array factor equations presented above. Calculation of exact element gain patterns or even the measurement of them is a difficult exercise. They can be estimated using MoM techniques but the technique requires the radiating element to represented by wires or patches and detailed knowledge the impedance of each in order to correctly calculate the current flows.

All literature reviewed in the course of this research, used either a simple theoretical element pattern such as '$\cos(\vartheta)$', or assumed the array was comprised of isotropic radiating elements which does not sufficiently predict the performance of real arrays.

It is important to include installed element patterns for two reasons - firstly, it maintains the validity of the chosen modelling method and secondly it demonstrates that any performance optimisation methods developed in the course of this research will remain valid when applied to real (i.e. non isotropic) antenna arrays.

Theoretical element gain patterns are well understood and often are provided in manufacturer's data sheets, but again, they are ideal patterns and assume that the element is radiating in isolation from other neighbouring elements and the influences of the antenna housing or platform are ignored.

As this work contains theoretical array concepts, a full accurate set of element embedded gain patterns is not available (this would require measurement of real hardware). In the absence of such information, a theoretical or representative gain pattern has been used as a basis for the embedded patterns, but made to be more realistic by distorting the patterns to simulate distortion due to coupling within the array.

Installed element gain patterns deviate from their ideal theoretical responses due to a number of factors:

- mutual coupling with neighbouring elements,
- edge effects (i.e. elements at the edge of the array have fewer neighbouring elements to couple with and so have different patterns to those in the centre of the array,
- coupling with the host platform or structure,
- errors due to dimensional and tolerancing experienced during manufacture.

Unless the manufacture of each radiating element is identical, it is fair to assume that each element pattern will be unique.

Unique element gain patterns can be simulated by taking a theoretical element gain pattern (such as a cosine response) and attenuating it at a number of random points. Such patterns are not necessarily representative of real installed patterns but they are sufficient in order to emulate at least to some degree, the unpredictable nature of installed element patterns when performing studies on theoretical array designs.

The complete procedure for the generation of unique element gain patterns developed for this work is to attenuate a theoretical response and then generate a polynomial approximation to it. The polynomial has the effect of smoothing sharp random attenuated points to give a more realistic pattern (see Figure 2-5).

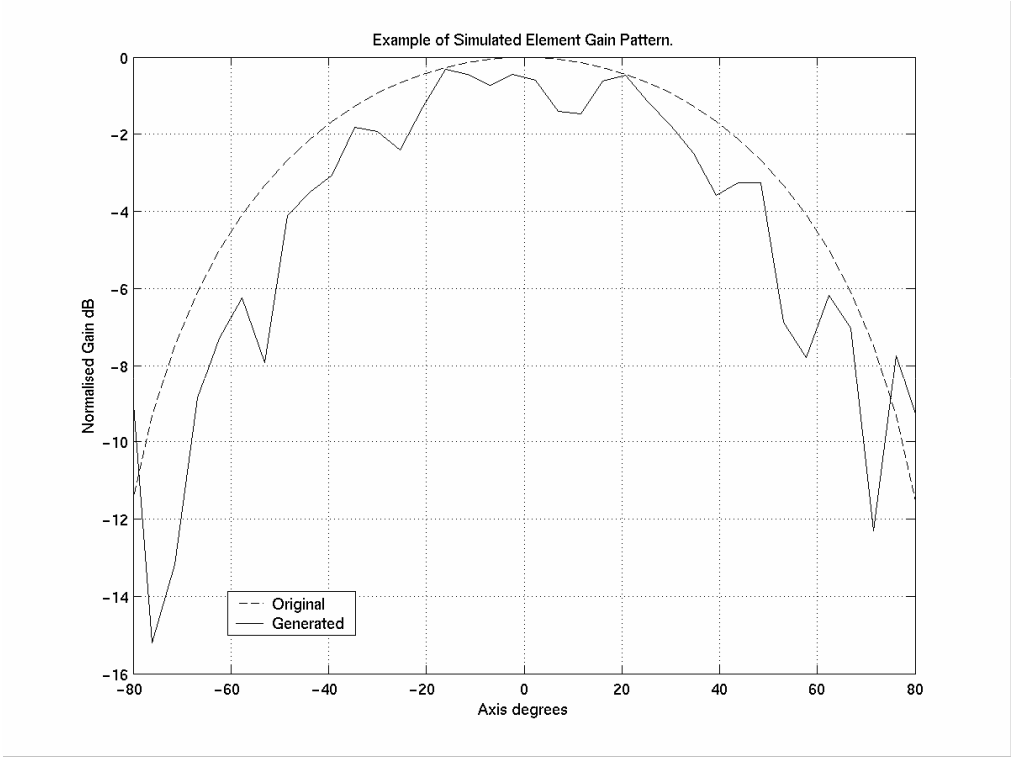


Figure 2-6 Example of generated element gain pattern.

When modelling a theoretical array design, a unique pattern can be created for each element in the array so as many as several thousand can be required. In practice they can be calculated once and stored in a look-up table during optimisation of performance. If real data became available, it could then be substituted and used in the same manner.

The embedded patterns of elements in real arrays behave quite differently at the edges of the array due to the inter-element coupling experienced. This effect is widely reported in the literature. In order to take this phenomenon into account and improve realism of the analysis, the random attenuation applied to the element patterns was increased in magnitude in some angular regions for the periphery elements.

2.6. Conventions and Definitions

A brief summary of antenna terms is given in Appendix A. that corresponds with the IEEE standard definition of terms for antennas [95]. These definitions have been adopted throughout this work.

CHAPTER 3
Antenna Model Validation

3. Antenna Model Validation

An important stage in any model development is the validation exercise. For the model to become a stage in a system optimisation algorithm, it must give realistic and accurate results.

To gain confidence in the performance of the antenna software models, a number of radiation patterns were generated. Features of the patterns were then compared with expected results from theory in a number of different tests.

In all, four different models have been developed and tested. They comprise of linear, planar, circular arc and conformal array models and are demonstrated below.

3.1. Linear and Planar Array Models

The first test consisted of generating a radiation plot for a sixty-four isotropic element (8 x 8) planar array and comparing the positions of the sidelobes with calculated values. The plot is shown in Figure 3-1.

The formula for calculating the positions of sidelobes generated by linear arrays is given by

$$\sin \theta_s = \pm \frac{(2q+1)\lambda}{2md_x} \qquad \text{Eq. 3-1}$$

where q is an integer value (1,2,3…), m is equal to the number of elements and d_x is the spacing between the elements. Equation 3-1 also applies to uniformly spaced planar arrays. Setting $m = 8$, $d_x = 0.015$, the sidelobe positions θ_S, are calculated as:

$$\theta_S = \pm 22.02431^\circ, \pm 38.68219^\circ, \pm 61.04498^\circ \ldots$$

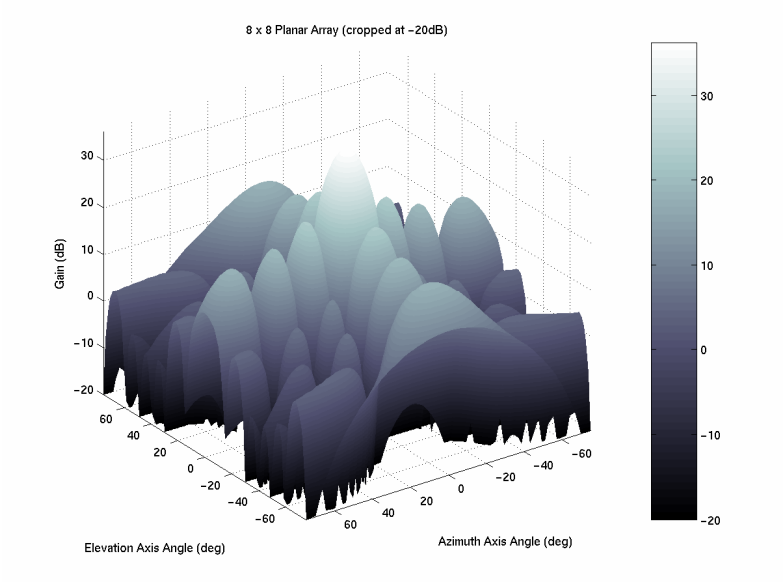


Figure 3-1 Radiation Plot of an Unsteered 8 x 8 Element Array.

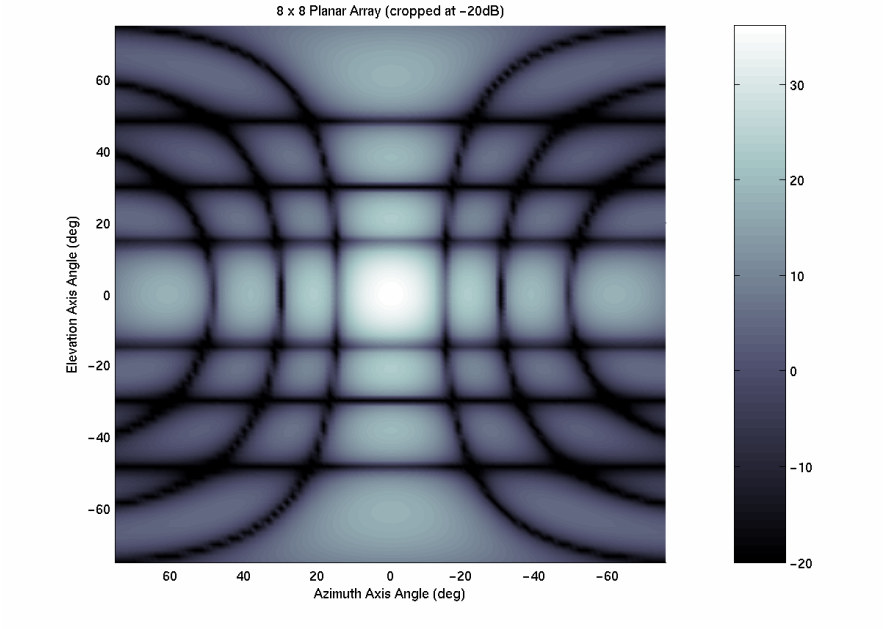


Figure 3-2 Contour Plot, 8 x 8 Array with Uniform Illumination

Analysis of the sampled data identified sidelobes at:

$$\theta_S = \pm 21.667^\circ, \pm 38.333^\circ, \pm 60.000^\circ \ldots$$

These values agree well with the sidelobe positions on the radiation pattern plot produced using the model (the errors are within the sampling resolution which was set at 1.667º in this case).

There is some widening of the pattern along the elevation dimension particularly noticeable at the far-out sidelobes. This widening is because the plots shown are Mercator projections of the radiation pattern as it is easier to read and analyse the plots. The effect is the same as is experienced when transferring a spherical globe map of the earth into the XY plane. The effect is particularly noticeable when the array is steered. The alternative is to use 3D-polar plots but it becomes difficult to directly read values from them.

The pattern in the first test uses a uniform excitation of the elements. In order to validate the model performance further, it was necessary to try some classical illumination functions such as a Chebyshev distribution and monitor the affect on the radiation patterns produced. The MATLAB 'chebwin' function was used to generate a Chebyshev amplitude distribution for a 20 element linear array.

Similarly a Chebyshev distribution for a 20 x 20 element planar array was calculated and used in the planar array model. The Chebyshev pattern was set to produce sidelobes at -30dB (relative to the mainlobe).

The software worked correctly and produced radiation patterns with sidelobes of equal magnitude at –30dB. The characteristic widening of the mainlobe (expected with Chebyshev distributions) was also evident in the plots produced by both the linear and planar models, as in Figure 3-3.

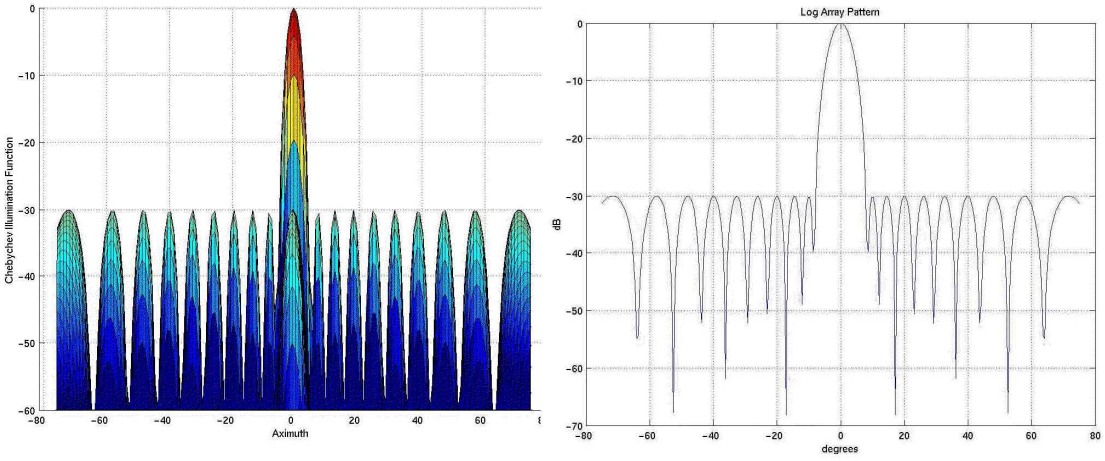


Figure 3-3 Test Chebyshev Patterns

(Left: planar model viewed from the azimuth direction, Right: linear model radiation pattern cut).

The sidelobe positions and expected magnitudes were faithfully reproduced in the radiation patterns.

The next test involved generation of a linear array power pattern cut, with uniform illumination. The first sidelobes occurred at –13.3dB which agrees with well-known theory [96]. The generated pattern can be seen in Figure 3-4.

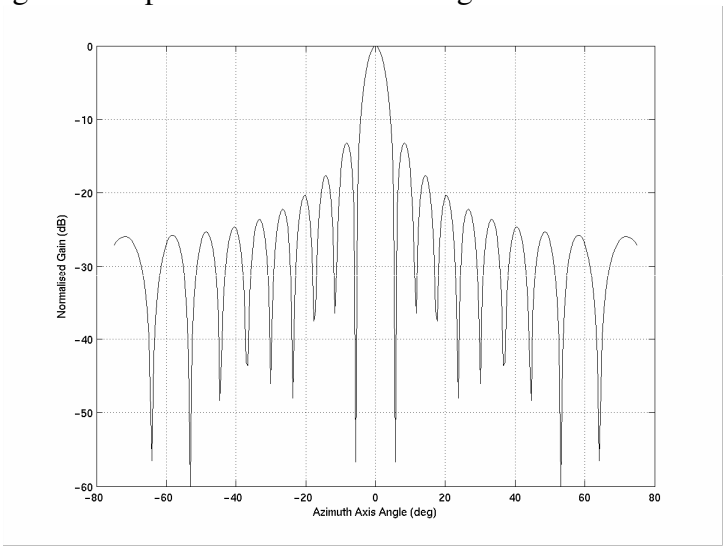


Figure 3-4 Validation of Model: Sidelobe Level Check.

To check that the model responds correctly to steered beams, a sweep of steering angles was performed. Figure 3-5 shows a pattern scanned to 35° in both azimuth and elevation (the image has been cropped at -20dB).

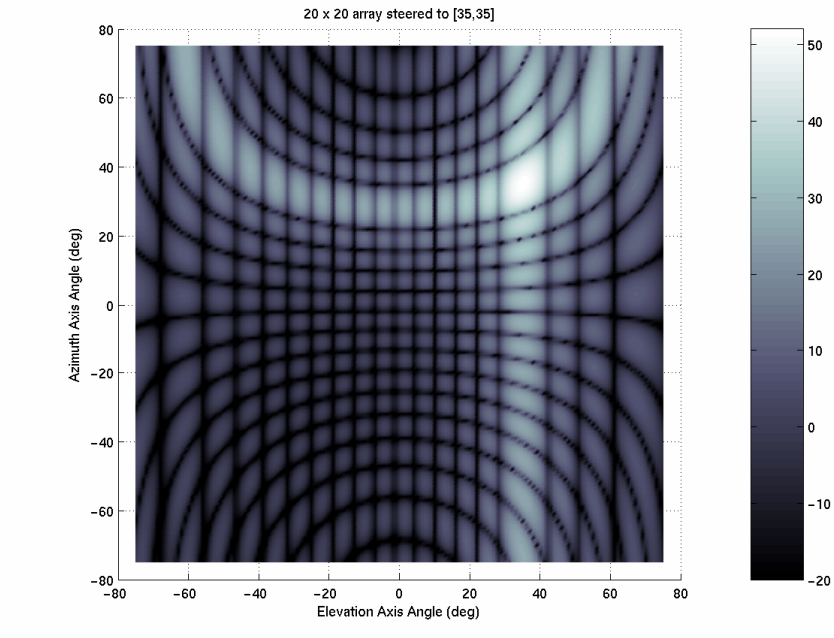


Figure 3-5 Validation of Model: Steered Beam Pattern.

The mainlobe position is correct, and the figure demonstrates the complexity of the sidelobe structure when the beam is steered.

3.2. Conformal Arc Model

This model determines the angular separation of the elements on the arc and then calculates the radiation pattern by evaluating the element factor and array factor at each sample point. One benefit of conformal arcs is that their maximum scan angle can be increased beyond the limits of a linear or planar array so the radiation pattern was calculated over ±120° rather than the ±75° used so far.

In order to validate the model performance it was necessary to find an example of a circular arc array synthesis in the literature and to compare results with those obtained using the code. Conformal arrays have not received the same attention in the literature as planar or linear antennas and suitable examples are harder to find.

One suitable reference was provided by Brann and Virga [97]. The complete amplitude and phase values for a 24 element circular arc array antenna were presented together with enough detail to permit a comparison exercise to be performed. In [95], the phase and amplitude sets were generated using a simple single-objective genetic algorithm. The element pattern was given by $\cos(\phi - n\Delta\phi)$ where ϕ is the azimuth component of the observation angle and n is the element number. The amplitude and phase values were presented in plots and have been reproduced here in Figure 3-6.

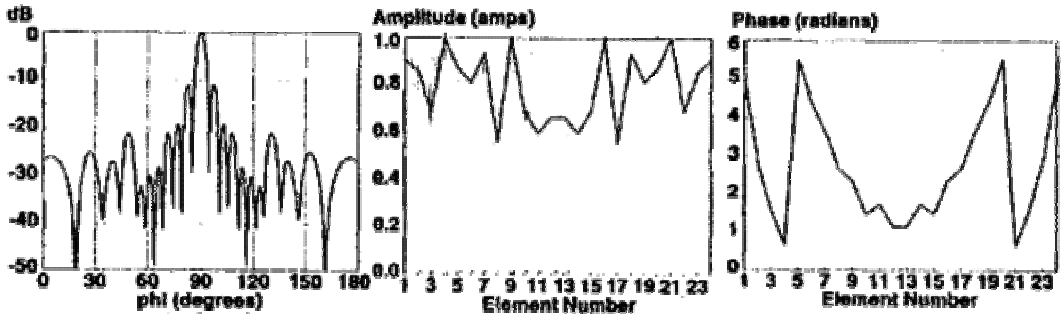
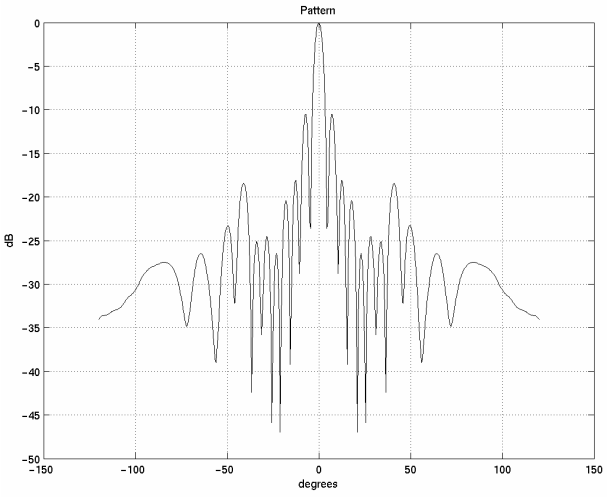


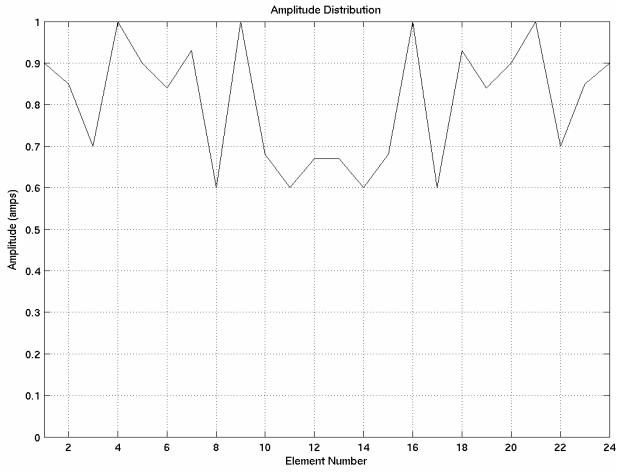
Figure 3-6 Conformal Array Model Sample Data.

The exact values were not given for the Bran and Virga array so the data was read from the plots and used in the Matlab code. All of the other details such as the array geometry used in the example were faithfully emulated and the radiation pattern produced by the model matches the published pattern well (Figure 3-7). This test provided confidence that the model was operating correctly.

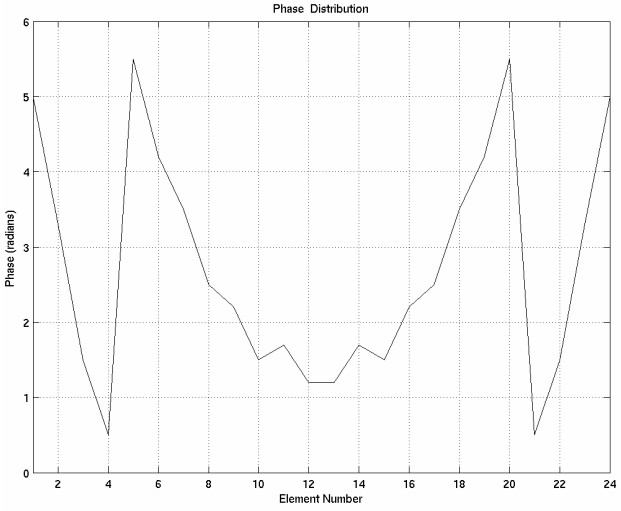
(a)



(b)



(c)



(a) Radiation Pattern (b) Amplitude distribution (c) Phase Distribution

Figure 3-7 Conformal Array Model Validation of Performance.

3.3. Accuracy

It should be noted that no single EM modelling technique can completely and accurately describe the radiation pattern of large array antennas and that some inaccuracies will result, whichever method is chosen.

The software models developed in this work sample the radiation pattern, and the accuracy of the final plot is dependent on the resolution of the samples. With efficient code it is possible to achieve 0.5˚ resolution within reasonable processing time. Little improvement can be made by further increasing the sampling resolution due to the Nyquist Sampling Criteria [98,99]. Interpolation is used by the plotting function to generate the smooth curves evident in the radiation pattern plots, but any later analysis of the radiation pattern characteristics are performed on the actual sampled values.

In any event, these models can be considered to provide good, first-order approximations of the antenna radiation patterns.

3.3.1. Quantisation Errors

With a n-bit digital phase shifter, the phase of one element can only be set to discrete values, for example 0, 45, 90,…....315 (3-bit phase shifter).

The quantisation introduces a beam steering error, quantisation sidelobes in the pattern and an increase in the average sidelobe level. Generally in large arrays that have more than 3 or 4 bits of phase control, the steering error is negligible and the quantisation lobes are rarely objectionable. The effect can be mitigated by placing a randomised fixed phase shift in each element's path and simply subtracting the phase shift from the required taper.

Figure 3-8 shows an ideal Dolph-Chebyshev synthesised pattern for a 51 element linear array. The pattern is set for uniform sidelobes at -30dB. The affects of generating the pattern using six-bit amplitude quantisation can be seen in Figure 3-9 and is negligible.

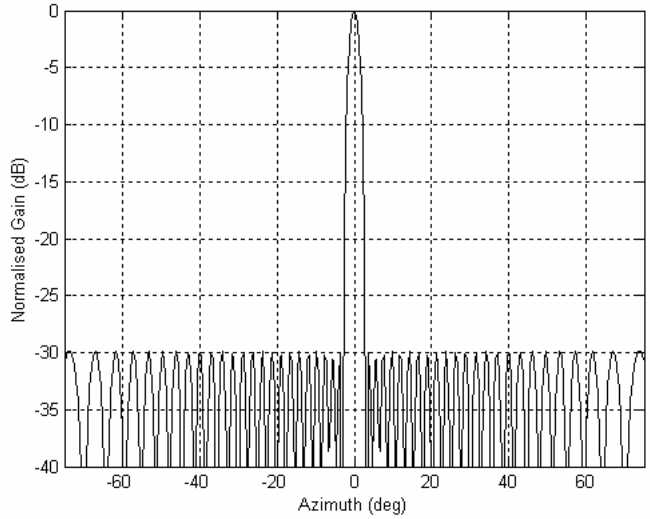


Figure 3-8 Ideal Dolph-Chebyshev pattern for a 51 element linear array.

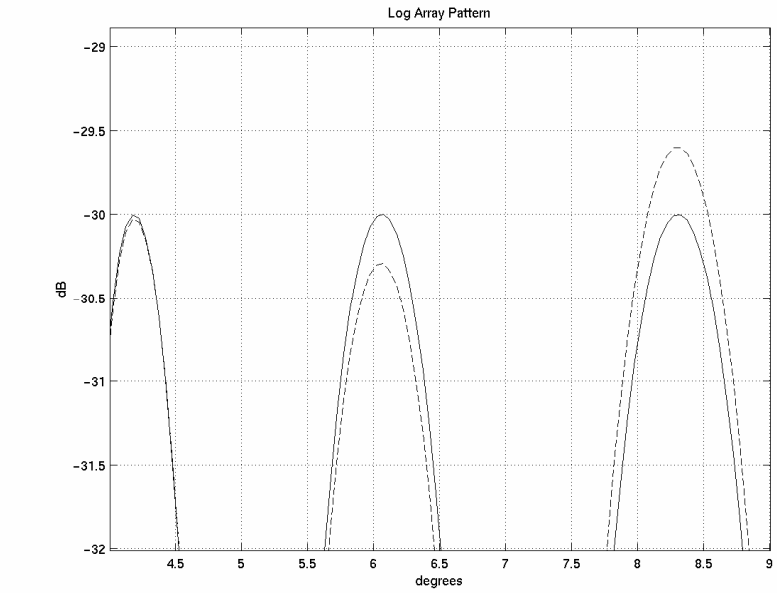


Figure 3-9 Dolph-Chebyshev Pattern.

Ideal Dolph-Chebyshev (solid line), quantised Dolph-Chebyshev (dotted line).

Skolnik has shown that it should be possible to obtain very low sidelobes using just 3 or 4 bit phase shifters in large array. Loss of mainlobe gain due to quantised phase shifters is only 0.23 dB for a three-bit phase shifter and 0.006 dB for a four-bit device. The antenna models developed for this work are capable of quantising both amplitude attenuation and phase control so that more realistic performance estimates are available.

CHAPTER 4
Evolutionary Algorithms

4. Evolutionary Algorithms

4.1. Optimisation Introduction

Optimisation, stated simply, is the process of finding the minimum or maximum of some mathematical function. An optimisation problem involving a single variable can usually be easily solved. Taking the derivative of an analytical function, setting it equal to zero and solving, results in the critical points of that function, including the optimal minimum and maximum values. Functions of two or more variables can be solved in a similar method. However, functions of multiple variables can prove very difficult to solve. When many hundreds or thousands of variables exist in a problem, classical optimisation methods are simply not powerful enough and new approaches are required.

Functions may also have multiple local maximum or minimum points. Some common optimisation techniques such as hill-climbing e.g. [100], have difficulty locating multiple maxima or minima. For example, consider a function with three maxima of equal value. In trying to locate the peaks of the function, standard hill-climbing techniques would start at some initial location and climb to the top of one of the peaks. This single peak would be declared as the maximum. Searching with this approach provides no information about the other peaks of equal value, nor does it prove anything about the global maximum. The global maxima could be located somewhere else entirely in the design space and the same problem exists when locating the minima. The choice of where to start the search has a large impact on the final answer when using gradient search methods.

It is possible to write optimisation algorithms that perform global, rather than local searches. One example of a global optimisation algorithm is the evolutionary algorithm (EA) [30,31]. Work by Mitchell and Holland [101] has shown that EAs can outperform hill-climbing techniques.

4.2. Evolutionary Algorithm Overview

This section and the remainder of this chapter contains an introduction to the basic evolutionary algorithm and outlines the procedures for solving problems using the simple genetic algorithm.

Evolutionary algorithms are becoming very popular with the electromagnetics community. The reason for the sudden popularity of EAs is simple - the gradient optimisation methods that are most popular in engineering disciplines have not performed consistently across the variety of EM design problems. The global search conducted by EAs is proving much more capable in this field of design.

Evolutionary algorithms are designed to search a much wider area of the design space, and could potentially provide a set of optimal solutions to a given problem. The EA approach was selected for this research because many different solutions can be expected to be found in antenna radiation pattern synthesis, and it is important for the antenna designer to explore as much of the potential decision space as possible, before selecting a single design or control solution. Often with complicated designs, an exhaustive search of the entire design space is not feasible due to the high computational burden. EAs can help find good solutions in a much shorter time.

The purpose of a EA is to search a problem's decision space in order to find its optimal solutions in objective space (Figure 4-1).

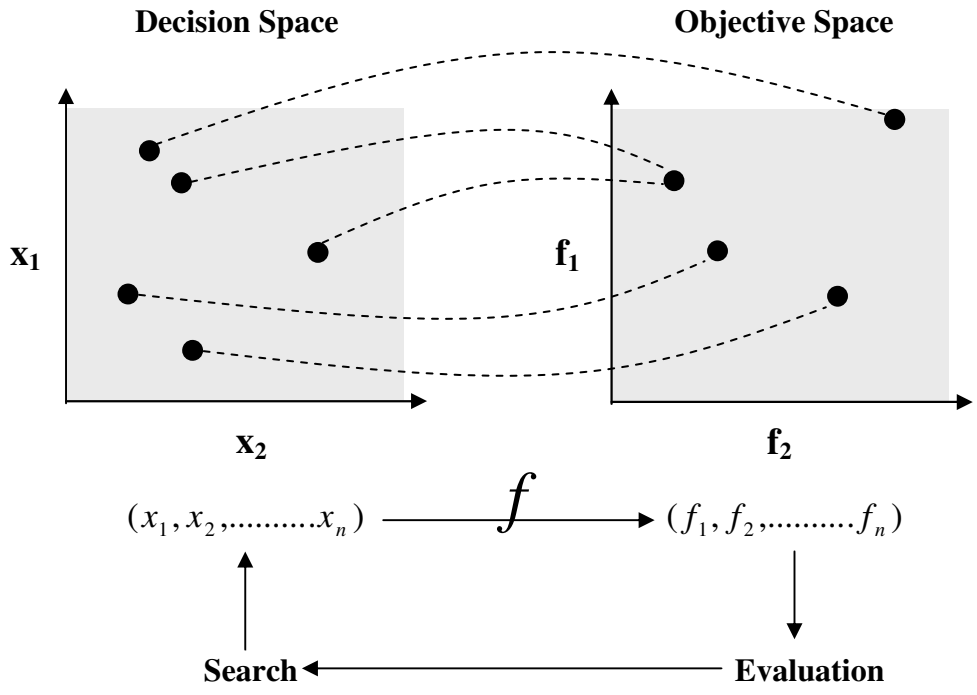


Figure 4-1 Decision and Objective Space

The EA is a stochastic global search method that mimics Charles Darwin's evolutionary theories of natural selection (survival of the fittest) [102]. EAs operate on a population of potential solutions, applying the principle of survival of the fittest to produce increasingly better approximations to a solution. In theory, it is possible for them to find true globally optimum solutions provided they exist within the decision search space and if certain optional genetic operators are included (see Section 4.4.3 on 'Mutuation'). At this point it is worth summarising some of the analogous terms, see Table 4-1.

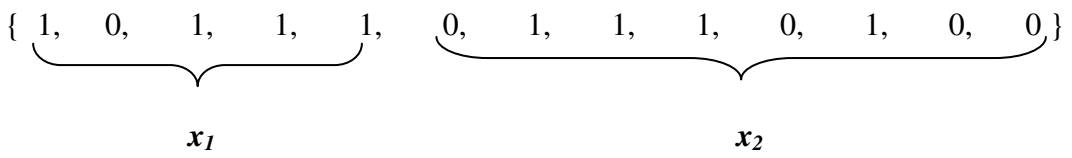
Optimisation Term	Evolutionary/Genetic Analogy
Variable	Gene
String or vector of variables	Chromosome
Set of variables that represent a single solution.	Individual (defined by one or more chromosomes)
Set of solutions	Population
Iteration	Generation
Success or performance	Fitness

Table 4-1 Evolutionary/Genetic Analogies.

At each generation (iteration), a new set of individuals is created by the process of selecting individuals (solutions) according to their level of fitness (success) in the problem domain and 'breeding' them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, in other words, better solutions to a problem.

4.2.1. Chromosomes and Genotypic Encoding

Individuals are encoded as chromosomes (strings), composed over some alphabet(s), so that the chromosome values known as genotypes, are uniquely mapped onto the decision variable or 'phenotypic' domain. The most commonly used representation in EAs is the binary alphabet {0, 1} although other representations can be used, e.g. integer, real-valued etc. For example, a problem with two variables, x_1 and x_2, may be mapped onto the chromosome structure as:



In this example, a binary chromosome has been used, where x_1 contains 5 bits and x_2 uses 8 bits. The number of bits used should be related to level of accuracy or the range of individual decision values that are required. A consequence of increasing the number of bits used to represent a parameter is the expansion of the decision search space size - the EA has to search a larger solution space and convergence can slow down. The increased size forces a problem specific trade-off between the probability of finding a globally optimal solution and the overall algorithm run time.

The EA search process is capable of operating on an encoding of the decision variables rather than the decision variables themselves. This encoding is where one of the main

strengths of an EA lies - the genotype need not directly contain numerical optimisation values and can instead contain complex encoding of systems, processes or methods, the result of which represent a solution to a problem. For now, we will just consider the encoding of simple binary strings but as this work progresses, more complex examples of genotypic encoding will be demonstrated, some of which is unique to this research (see Chapter 5).

The chromosome can contain real values that directly represent variables in the problem, in which case the search is directly in the phenotypic domain. A binary encoded chromosome requires decoding into the phenotypic domain.

The final choice of the chromosome-encoding scheme is left to the user and is by no means fixed for each problem type. A general rule is to keep the chromosomes as short as is possible to fully define a solution and remember that an individual solution may consist of several chromosomes each with different genotypic encoding.

4.2.2. Creating an Initial Population

After the chromosome encoding has been defined, the next stage is to generate a set of random chromosomes. These will form the initial population of individuals that will be evaluated and subject to the evolutionary optimisation.

The size of the initial population is a user defined parameter and should be decided upon with reference to the number of variables to be optimised and the total number of solutions in the decision space.

Too small a population and the search may not be efficient, too large and the algorithm may not converge. The chromosomes in later generations will largely be formed using the genes contained in the initial population and so the diversity of the initial 'building blocks' can influence the exploration of the search space [103].

Population sizes of 30, 60 or 100 are common, but some researchers use population sizes of several hundred or more. The final choice is often decided by time taken to evaluate a single solution. There are variants of EAs known as Micro-Genetic Algorithms that use a very small population size of around 10 individuals in order to speed up convergence and attempt to operate in real-time applications [104].

4.2.3. Objective Functions and Fitness Assessment

Once the chromosome(s) belonging to each individual in the population have been decoded into the phenotypic domain, it is possible to assess the performance, or fitness, of each one.

This assessment is calculated through an objective function that characterises an individual's performance in the problem domain. In the natural world, the performance would be an individual's ability to survive in its present environment. In antenna performance optimisation, the objective function may take the form of a combination of the antenna models and analysis techniques described in the last few chapters.

Evolutionary algorithms rely on many evaluations of the objective function to guide their search and so it is important that the functions are as efficient as possible. To illustrate the point, consider an EA with a population size of thirty that is allowed to evolve for 60 generations. There will be 1800 fitness function evaluations. If the EA takes one second to evaluate the fitness of each solution, the total run would take over half an hour.

Each individual will be given a fitness value derived from its raw performance measures calculated using the objective function. For example, in a maximisation problem, a better (fitter) individual will have a higher fitness value than a weaker solution.

In antenna optimisation, we often have more than one parameter to optimise as shown in the last chapter (see Section 2.4). Multiple objectives are very common in optimisation, as rarely are we interested in just one objective value. Evaluation of a candidate solution could require many different objective functions to be calculated.

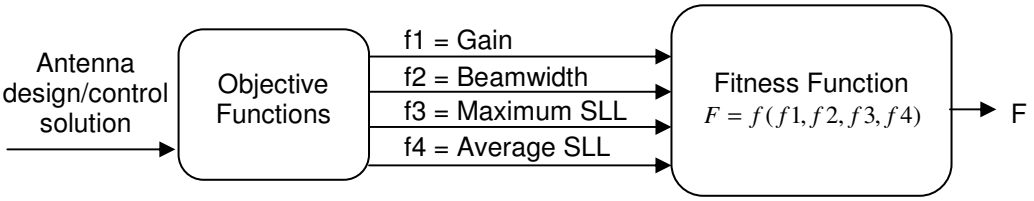


Figure 4-2 Example of Objective/Fitness Function Evaluation

EAs are guided by a single fitness value so the objectives have to be combined in some way. The combination is achieved by using a fitness function that is simply some function of the objective values. Common fitness functions include sums of objectives (Eq. 4-1) or weighted sums (Eq. 4-2).

$$F = f_1 + f_2 + f_3 .... + f_n \qquad \text{Eq. 4-1}$$

$$F = w_1 f_1 + w_2 f_2 + w_3 f_3 .... + w_n f_n \qquad \text{Eq. 4-2}$$

There are many variations of fitness function and some degree of experimentation is required to determine one most suitable for a particular problem type. This assignment of fitness values establishes the basis for selection of pairs of individuals that will be mated together during reproduction. In order to keep the analogy with the process of natural selection, the fitter solutions must be selected for reproduction more often than the weaker solutions.

4.3. Selection Mechanisms

Those individuals with high fitness values (relative to the spread within the population) have a high probability of being selected as parent solutions. Poorly performing solutions are not normally excluded from the selection process, but are far less likely to be selected as parents. Typically, two individuals will be selected and their chromosomes recombined to produce new offspring solutions that will be evaluated in the next generation. The number of new (child) solutions that are produced need not be equal to the current population size, although it usually is. See section 4.5 for more discussion on this issue.

Having calculated a fitness value for each individual in the population, a number of different selection strategies are available and again, it is down to the user to choose one that performs well in the problem domain. Two examples of selection mechanisms are given below.

4.3.1. Proportional Selection

Proportional selection biases the selection towards individuals with higher fitness values. A common method is the so-called 'Roulette Wheel Selection'. The circumference of the wheel is equal to sum of all the individual fitness values, and wheel is divided into section proportional to each individual's fitness. Figure 4-3 shows an example of the proportional representation for a selection pool of ten individuals.

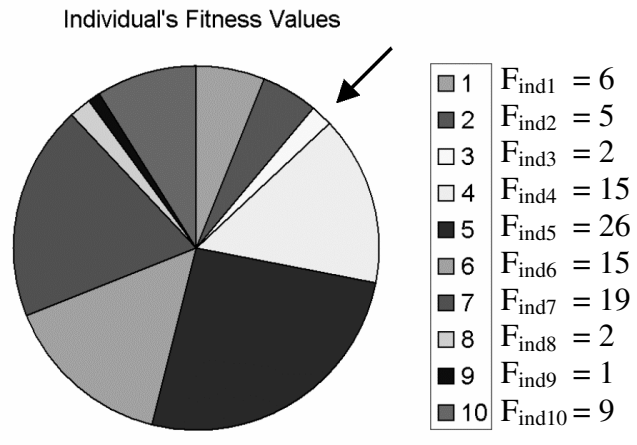


Figure 4-3 Roulette Wheel Selection

4.3.2. Tournament Selection

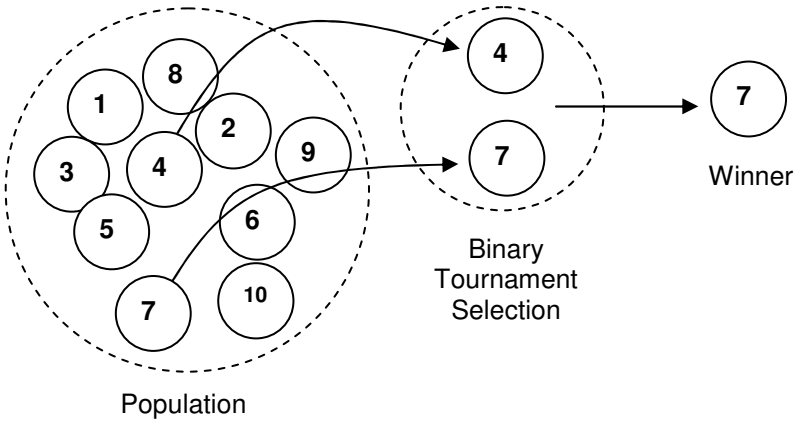


Figure 4-4 Binary Tournament Selection

It is not necessary to have all the population members in the selection competition. For example, 'binary tournament selection' will use just two individuals chosen at random from the population. This method can help introduce diversity in the search by preventing the selection being dominated by just one or two individuals with high fitness values.

Figure 4-4 shows an illustration of binary tournament selection (fitness values are as in Figure 4-3). Two individuals are chosen at random from the population to form the competitors in a tournament selection. When maximising, the solution with the highest fitness value becomes the contest winner and is selected as a parent. The procedure is repeated if two parents are required to form offspring solutions (the number of parents required is dependent on the exact formulation of the EA - an offspring solution could be produced by just one parent).

Both Roulette Wheel and Tournament Selection are operators commonly used in the Genetic Algorithm.

4.4. Genetic Operators

Once two individuals have been selected as parents, a number of genetic operators are applied to their chromosomes to form new offspring.

The basic genetic operator is known as crossover or recombination. Like its counterpart in nature, crossover produces new individuals that have some parts of both parents' genetic material.

4.4.1. Single/Multi-point Crossover

A basic type of crossover is known as ‘single-point’. A single-point crossover forces a break in the chromosomes of the parents so that each child obtains genetic information from each parent. In order to keep the population size constant, two children are produced from two parents.

The break is made randomly and one child gets the binary code of one parent to the left of the break, while the binary code to the right of the break comes from the other parent. The other child gets the opposite. Each child inherits certain traits from both parents in this manner. The procedure is best explained diagrammatically:

Single-point break

Parent 1	**11011	00100110110**
Parent 2	01010	11000011110

Child 1	**11011**	11000011110
Child 2	01010	**00100110110**

In the chromosome shown, each gene has two possible values, 0 and 1 (the set of valid values for each gene is known as the alleles). There are other types of crossover - multi-point crossover allows multiple breaks in the chromosome instead of a single break and uniform crossover gives a 50% chance of each allele coming from either parent. The disruptive nature of multi-point crossover appears to encourage the exploration of the search space, rather than favouring the convergence to highly fit individuals early in the search, thus making the search more robust [105]. Both single and multi-point schemes are equally applicable to both real valued and binary chromosomes. Many studies have demonstrated that single point crossover, although simple, does not perform well as the first and last genes can never be passed together to a child solution [106].

4.4.2. Uniform Crossover

Uniform crossover [107] is a simple recombination scheme that makes every bit or value a potential crossover point. A binary string the same length as the chromosome structures is created at random and the parity of the bits in the string indicates which parent will supply the offspring with which bits. Consider the following two parents P1 and P2, binary string S, and resulting offspring C1 and C2:

P1	=	1	1	0	0	1	1	0	1	0	1
P2	=	1	0	1	0	1	1	1	0	1	0
S	=	1	0	0	1	1	0	0	0	1	1
C1	=	1	1	0	0	1	1	0	1	1	0
C2	=	1	0	1	0	1	1	1	0	0	1

Spears and De Jong [108] have demonstrated how uniform crossover may be parameterised by applying a probability to the swapping of bits. This extra parameter can be used to control the amount of disruption during recombination without introducing a bias towards the length of the representation used. When uniform crossover is used with real-valued genes, it is usually referred to as discrete recombination.

4.4.3. Mutation

After selection and crossover, mutations are also permitted in order to explore regions of the design space that may have already become extinct or have never been explored.

In Genetic Algorithms, mutation is randomly applied with low probability, typically in the range 0.001 and 0.05, and modifies elements in the chromosomes. Mutation is often seen as a mechanism for ensuring the probability of searching any given string will never be zero. It also acts as a safety net to recover good genetic material that may be lost through the action of selection and crossover [30]. Each new child solution is a candidate for mutation.

A creep mutation randomly selects a single bit to be changed (child 1) and a jump mutation swaps two random bits within the child's binary string (child 2):

Child 1	110**1**1110000011111
Mutated Child 1	110**0**1110000011111

Child 2	0101100100110110
Mutated Child 2	0101101100110100

Mutation is to be used cautiously as it can prevent population convergence (see Section 4.5.1) if it is applied too often. With real-valued encoding, a mathematical operation is performed on values within the chromosome. The operation may be a simple multiplication or division and again, there are many different schemes available. For example one scheme may take a single value and change it to the maximum (or minimum) possible according to the range bounds for the variable. Others are subtler and may apply only a small change to the variable value.

Wright [109] demonstrates how real-coded GAs may take advantage of higher mutation rates than binary-coded GAs, increasing the level of possible exploration of the search space without adversely affecting the convergence characteristics. Similarly, Tate and Smith [110] argue that for any genotypic codings containing alphabets more complex than binary, high mutation rates can be both desirable and necessary. They give examples of how high mutation rates and non-binary coding yielded significantly better solutions than the normal, more conservative approach to mutation.

The exact mathematical operation (or inversion of bits) used is best chosen with regard to the type of chromosome encoding scheme used and the size of the search space.

4.5. Reinsertion and Elitism

It is common to see fixed population sizes in EAs as they are easy to implement. Typically, the original population is completely replaced by the new offspring solutions and the EA is described as being steady-state [111].

If fewer individuals are produced by recombination than the size of the original population, then the fractional difference between the new and old population sizes is termed a generation gap [112]. If one or more of the highest fitness individuals are deterministically allowed to propagate through successive generations, the GA is said to use an elitist strategy.

Elitist strategies compare the fitness values of a new generation with those of the previous generation. If the highest solution from the previous generation is higher than the best solution from the new generation, an individual from the new generation is removed and the previous best is inserted to replace it. This ensures the survival of the fittest rule applies between generations as well as within them. Haupt [56] ensured elitism occurred by keeping the top 50% of each population during each generation, but this approach does not make for an efficient search as their will be fewer new chromosomes contributing to the search in each generation.

Elitism is a useful function, particularly when search spaces are large and good solutions prove difficult to find and maintain in the population.

4.5.1. Termination

There are no set rules for termination of an evolutionary algorithm. Termination criteria may be set at some given number of generations, or after some measure of convergence has been reached. Eventually, the population will tend to converge to a common point. An illustration of this convergence can be seen in human biological evolution. The Europeans and Africans each progressed down separate paths. The Europeans developed pale skin, while the Africans developed dark skin. These isolated populations are said to have converged because each individual, within the separated populations, holds a common trait. Technically, while Europeans lost the dark pigments from their skin, Africans maintained it – it helps to prevent sunburn and also reduces production of vitamin D.

The EA, as an optimisation tool, can also arrive at this kind of convergence in design within its encoded design parameters.

In practice, a standard GA should be run several times to account for this convergence and the inherent random processes. The initial creation of individuals, selection of parents, crossover reproduction, and child mutations are all based on random number draws. Different results can be expected between one initial randomisation seed and another. On the other hand, these differences are not guaranteed and different seeds could end with the same results. It is good practice to record the seeds used to initialise random number generators so that the results found are repeatable.

It is common to simply stop a GA once a certain number of generations have been completed. Stopping the GA raises the question “Has the GA found the best possible solution to my problem?” and unfortunately, the only way to answer the question is to perform an exhaustive search. A good practice to follow when deciding upon termination of the algorithm is to ask the question “Has the GA found a good solution?”.

4.6. GA Flow Chart

The diagram shown in Figure 4-5 shows the each of the steps in a GA. When applying a GA to a new problem, it is necessary to fine-tune some of the important settings in order to improve the overall performance. The number of individuals in a population is one such parameter. Too few individuals will restrict the search while too many will slow it down. The population size is usually balanced against acceptable run times. The other settings that influence performance are the probabilities of occurrence of the genetic operators (crossover and mutation). Crossover and mutation occur with distinct probabilities defined at the start of the algorithm. Probability of crossover is usually high as crossover is the main mechanism for exploration, set typically at 60 to 80%. If crossover does not occur, the offspring are created as exact copies of their parents. Probability of mutation is much lower as it can disrupt convergence. Typical values are 1% to 5%. (Note 100% mutation level turns the algorithm into a random search).

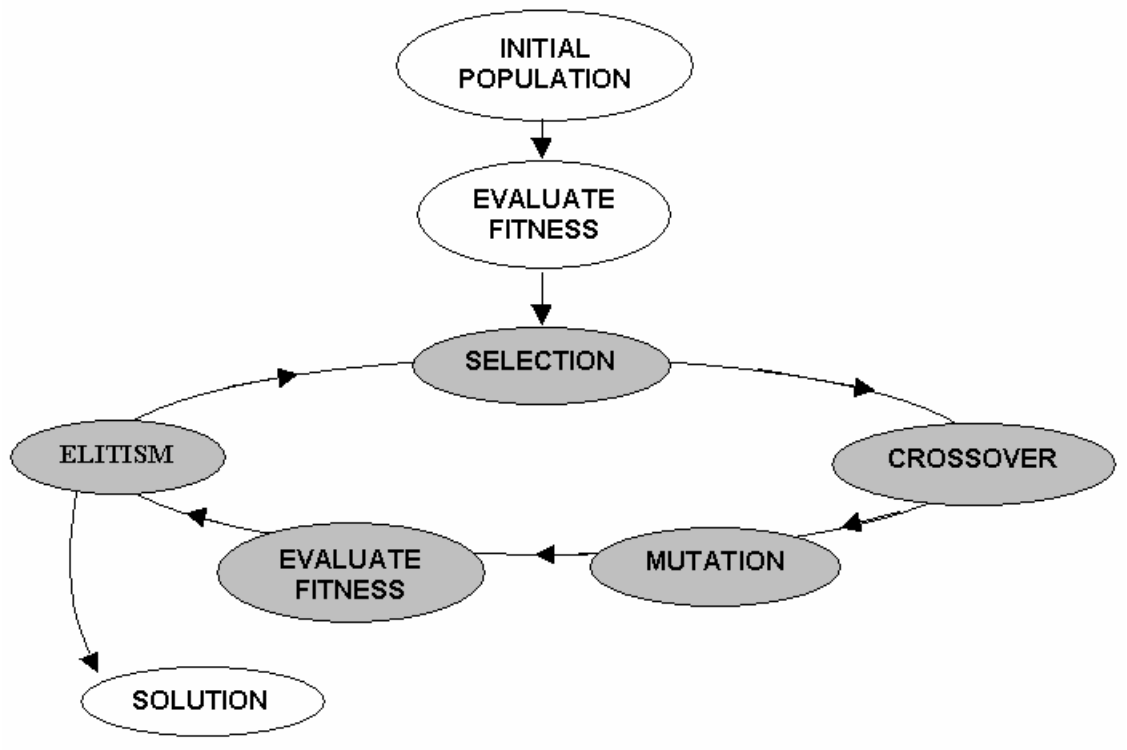


Figure 4-5 GA Flow Chart

This chapter has described how varied the EA can be in both its selection processes and in its genetic operators. The next chapter explores these options in more detail in order to produce a well performing GA design for the optimisation of antenna performance.

CHAPTER 5
Antenna Array Optimisation Using Genetic Algorithms

5. Antenna Array Optimisation Using Genetic Algorithms

5.1. Introduction

One of the key benefits of active array antenna technology aside from their increased reliability and beam agility, is the ability to dynamically shape and control their radiation patterns. This additional control can be of great advantage as it allows arrays to be used in new ways. For example, beam positions can be switched at high speeds enabling new types of scan patterns to be followed, or the antenna can be tasked to interleave several different functions in a time-shared manner.

The simplest form of array is the linear array. Linear arrays are widely used in communication, navigation and identification sensors. One application of the linear array antenna is in Identify Friend or Foe equipment (IFF) which typically uses four separate arrays each scanning ±45° to achieve full coverage in azimuth.

Consider a simple, uniformly spaced linear array that contains twenty isotropic elements mounted on a reflective backplane to prevent radiation in the rear direction, as in Figure 5-1.

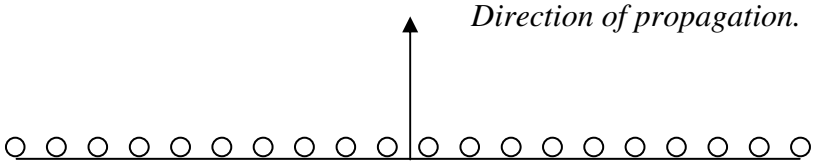


Figure 5-1 Twenty Element Linear Array.

Each element in the array will require a phase (α_n) and amplitude setting (a_n) in order to control the power and direction of the transmitted energy (hence the radiation pattern). In real systems, these values are controlled digitally and must be quantised. If there were 6 bits (i.e. 64 increments) of phase control and 9 bits (i.e. 512 increments) of

amplitude control in a state of the art system, there would be 32768 (64 x 512) possible combinations of amplitude and phase for a single element. For the entire array of just twenty elements, there are a massive 2.037×10^{90} complex excitation sets possible. The high number of possible excitation sets brings with it the advantage of providing an enormous amount of scope for controlling the array's radiation pattern.

The phase values can be fixed when considering a particular beamsteering angle, and this reduces the number of excitation sets in the example above to 1.53×10^{54}. The figure is still very large, and makes exhaustive searching for a desired radiation pattern impractical.

As discussed in Chapter 1, classical means of weighting array excitations tend to solve a very specific problem type such as finding the narrowest beamwidth for a specified sidelobe level (e.g. Dolph-Chebyshev [4] or Taylor [5]).

Dolph-Chebyshev illumination functions are often used as they produce patterns with the lowest beamwidth possible for a given sidelobe level. The beamwidth is defined here as the distance between the first nulls either side of the main beam. One defining feature of the pattern is that the sidelobes are all of equal magnitude. Although an illumination function of this type seems attractive (particularly for radar operation), it is seldom used in practice. The function is rarely used because as the array size increases, the currents at the end of the array become large compared with those in the rest of the aperture, and the radiation pattern becomes sensitive to the edge excitation. The high currents set a practical upper limit to the size of the array that can use the weighting function and also a lower limit on the achievable beamwidth.

The Taylor distribution is more achievable and approximates a Dolph-Chebyshev pattern in the vicinity of the main beam only. Outside of this uniform sidelobe region, the sidelobes progressively reduce in magnitude. The main disadvantage with the Taylor distribution is that the beamwidth is larger than that achieved by the Dolph pattern.

Both the Dolph-Chebyshev and Taylor assume that the array is working perfectly and the element patterns are identical. Both these conditions can never be met in a real array. The functions represent a minute proportion of the possible number excitation sets available. To fully exploit an array's capabilities, new means of finding excitation sets are required.

Evolutionary Algorithms (Genetic Algorithms in particular) have been used as a means of searching the large set of excitation sets to find those that result in good radiation pattern performance, and examples of these were shown in Chapter one. This chapter investigates and develops the implementation of genetic algorithms in array performance.

5.2. Implementation

In the genetic algorithm (GA), the amplitude and phase values for a trial solution can easily be represented in chromosomes. The main purpose of this chapter is to look at the application of the GA and to determine good practice when it is applied to antenna

optimisation problems. As shown in chapter four, there are many different variations of the simple GA and it is important to determine if any of these variations improves the results of their application in antenna optimisation. This work is intended to provide a good foundation for later studies involving more complex GAs applied to larger, more difficult problems.

Regardless of the type of optimisation algorithm used, a model of antenna performance is required. Chapters two and three looked at antenna modelling and established a method of modelling antenna array performance. Chapter two also introduced methods for analysis of radiation patterns in order to determine how well a particular pattern meets our performance expectations.

At this stage, with the performance and assessment models complete, the system is ready for optimisation. In its most generic representation, the GA 'wraps around' this system model as shown in Figure 5-2.

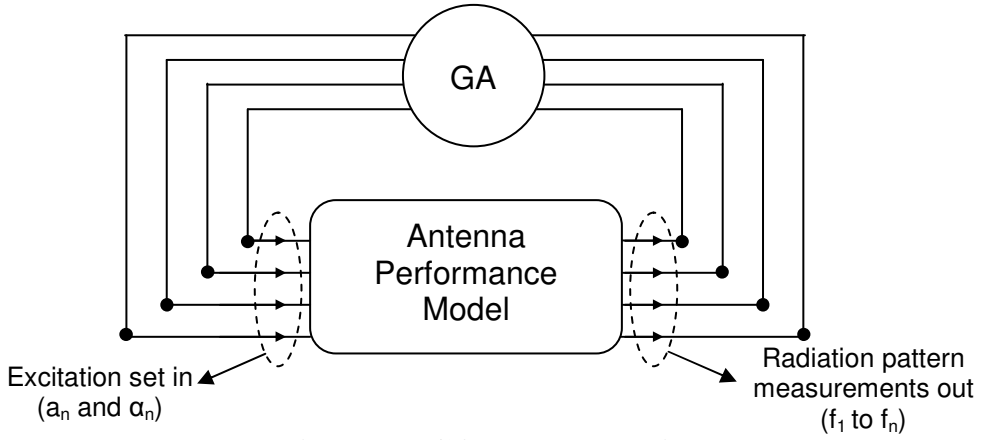


Figure 5-2 GA Implementation.

1. Step one is to randomly generate a population set containing a number of chromosomes that represent trial excitation sets.

2. Each chromosome (trial solution) is decoded into the real values of amplitude and phase, and is input into the antenna performance model. The model then calculates a radiation pattern.

3. Each radiation pattern is analysed and assigned objective values (f_1, f_2, … f_n) according the trial solution's degree of success in achieving each of a set of desired characteristics such as sidelobe level or beamwidth.

4. The objective values are then combined in some way to give a single overall measure of success known as the fitness value (F) for each solution. The fitness value is usually calculated by a straight summing or by using weighted sums. e.g.

$$F_1 = w_1.f_1 + w_2.f_2 \ldots\ldots\ldots\ldots w_n.f_n \qquad \textbf{Eq. 5-1}$$

   where w_1 to w_n are user defined weights.

5. Selection contests would then take place to determine the parents of the next generation of solutions and the standard genetic operators of crossover and mutation would then be used to generate the child solutions that will form the next generation.

6. Once a the population has been replaced by new child solutions, the procedure repeats from step 2, until some measure of convergence has been met, or a set number of generations (iterations) have past.

5.3. Genotypic Encoding Investigation

The amplitude and phase components of an element's excitation can be described by a complex number, with the real part containing the amplitude and the imaginary part containing the phase weighting.

The amplitude setting is usually normalised to a range of values between zero and one, and the phase value is given by an angular value in radians. In order to optimise an excitation set using a GA as described in the process above, the first question that has to be addressed is the type of alphabet to use in the genotypic encoding of the excitations.

While the weighting of the array is given by numeric values, the hardware implementation of these weights is invariably digital, and therefore quantised. This quantisation would imply that either a binary or real valued chromosome could be used to store the excitation set. The binary chromosome would ensure that all solutions searched in decision space are achievable solutions as the quantisation can be matched to the hardware, but the question has to be asked 'are there any advantages in using a real valued chromosome during the search, and then quantising the results?'. Most of the earlier cited researchers who applied genetic algorithms to this type of problem used binary encoding.

A simple genetic algorithm was set up to minimise the maximum sidelobe level of a 10GHz linear antenna array. At 10GHz, the spacing between the elements was fixed at 1.5cm (half wavelength spacing). The GA was run ten times using a binary chromosome, and a further ten times using a real valued chromosome (multiple runs were completed to account for the randomness and stochastic nature of the GA). The results were stored after each run.

The array contained thirty radiating elements and in all runs, the probability of crossover was set at 90%, probability of mutation at 5% and the population contained two hundred solutions. The GA was allowed to run for three hundred generations. The fitness value of each individual was given by:

$$f_n = G_{ml} - G_{msl} \qquad \textbf{Eq. 5-2}$$

where G_{ml} is the mainlobe gain (in dBs) and G_{msl} is the magnitude of the highest sidelobe found within ± 75º of the mainlobe (in dBs).

Table 5-1 below, shows the average and best solution found during each run, together with the average number of generations taken to find the best solution in each run.

	Max Fitness	Average Fitness	Average No. of generations before convergence
Binary Coded	20.1	19.55	163
Real Coded	19.55	19.16	108

Table 5-1 Real and Binary Genotype Comparison

The binary encoded GA managed to achieve sidelobes of magnitude -20.1 dB from the mainlobe and over the ten runs, achieved consistently similar results with little spread in the solutions found. Generally, the first fifty generations of search drew close to the final solution, will only minor improvements in the later generations. In some runs, a better solution was found in later generations after long periods with no improvement. These improvements were found to be due to the mutation operator modifying a variable. This particular GA converged (on average) after 162 generations indicating that the maximum number of generations (300) was excessively high. To avoid loss of gain, each excitation set found by the GA was normalised.

The real-coded GA achieved very similar performance, and on average, there was very little difference compared with the binary GA's results. The binary GA found the best solution, but it was only 0.5dB better than the best found by the real-coded GA.

The real-coded GA tended to converge more quickly than the binary (after 108 generations on average), and the earlier convergence can be viewed as advantageous. Once the real valued solutions were quantised the performance difference was minimal (typically less than +/- 0.2 dB).

Work by the earlier cited Tate and Smith [105], suggested that higher mutation rates could be beneficial with real valued chromosomes. In order to see if this finding is true and applies here, a number of real-valued runs were completed with different mutation values. The mutation operator simply replaced a single value within the chromosome with a random number. Mutation rates of 10, 20 and 50% were investigated and it was found that increasing mutation probability offered no improvement compared with the 5% level.

This example highlighted that the performance of a GA can be very problem specific and modifications to the genetic operators on one type of problem, may not improve the same GAs performance on other problems. The experimental work suggests that real valued chromosomes converge more quickly than binary encoded chromosomes on this type of problem. In addition, despite this earlier convergence, the quality of solutions found is as good as those found by the binary encoded GAs.

5.3.1. Improving Performance

Theoretically, when considering the same thirty element linear array, Taylor type distributions exist within the decision space that would deliver radiation patterns with -30dB or -40dB sidelobes (i.e. $f = 30.0$ or = 40). In order to investigate the robustness of a radiation pattern to minor changes in the excitation set, a -30dB Taylor weighting was calculated and applied to the array.

It was found that by changing just four out of the thirty excitation values within the set of -30dB Taylor weights, the magnitude of the sidelobes can easily be raised by 10dB or more. Varying a single excitation can raise the sidelobe level by up to 4dBs. This experiment highlighted that radiation pattern sets are sensitive to relatively small changes in excitation, and this fact can be taken into account when considering the search process. This fact suggests that once an optimiser appears to have converged, it may be beneficial to perform a second optimisation on a small number of excitation values.

In the GA used above, many solutions were found that achieved sidelobes of magnitude of -20dB, so it follows that changing a small number of excitations within one of these solutions may bring it closer to the performance of a -30dB Taylor type weighting.

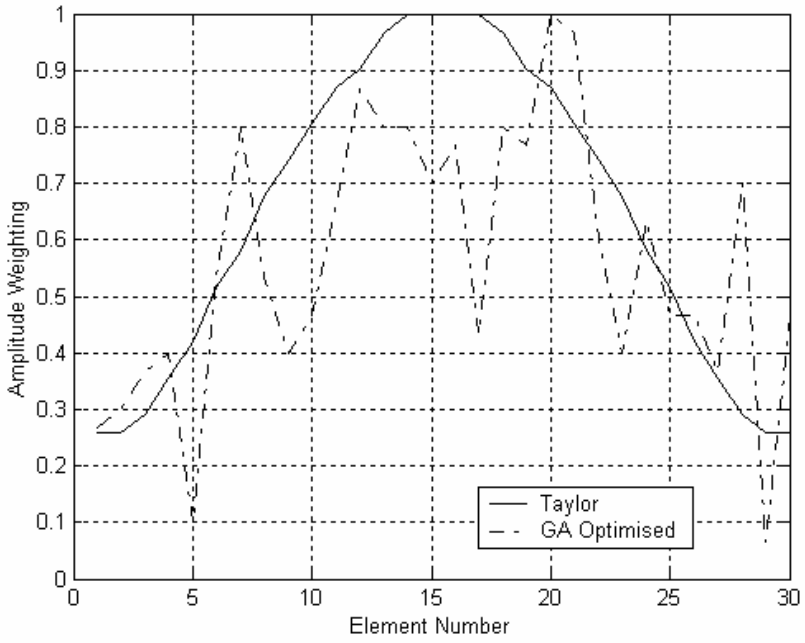


Figure 5-3 Comparison of a Quantised Taylor Weighting and GA Result.

A solution was chosen at random from the real valued runs above and its excitation set compared with a Taylor weighting. The two sets are shown in Figure 5-3. It can be seen that some of the excitation values are close to the Taylor weighting, and a smaller number of the excitations in the GA solution require modifying to bring them closer to the performance of the Taylor series.

A local search of the GA solution was used to identify which of the excitations required changing and by how much to increase or decrease their weightings. A second GA was set up to implement a localised search.

The genotypic encoding of this solution was achieved by using three chromosomes. The first, ch_1, contained ten random numbers between zero and one. This chromosome was used to represent how many excitations to vary within the set. The chromosome was decoded by counting the number of individual values within it that were greater

than 0.5, and the total was used as the number of excitations to be subject to change. for example if ch_1 contained the values:

$$ch_1 = [0.34, 0.677, 0.834, 0.022, 0.457, 0.199, 0.989, 0.233, 0.149, 0.245]$$

then it can be seen that three of the values are greater than 0.5, therefore three excitation values would be selected.

Similarly, a second chromosome ch_2, contained thirty random numbers between zero and one. This chromosome was used to determine the positions of the elements in the array that will be selected for excitation changes. The chromosome was decoded by referencing the initial positions of each random number and then sorting the random numbers in the chromosome from lowest to highest. This process is shown below:

Index	[	1,	2,	3,	4,	5]
ch₂	[	0.34,	0.91,	0.46,	0.1,	0.95]
Index	[	4,	1,	3,	2,	5]
ch₂ (sorted)	[	0.1,	0.34,	0.46,	0.91,	0.95]

This method proved to be a useful and efficient means of representing an index in a genotype.

A third chromosome ch_3 contained ten amplitude weightings, some of which may not be needed dependent on the outcome of decoding ch_1. It is important when there are potentially unused values within the chromosome to use a uniform crossover operator to prevent unused portions of the chromosome influencing the search (as they do not contribute to the success of a solution).

The purpose of encoding the three chromosomes in this way was to maintain their validity once the genetic operators of crossover and mutation have been applied. This encoding shows the versatility of the GA and illustrates how chromosomes can be used to represent structures or sequences of events, rather than being limited to specific optimisation variables such as amplitude weightings.

A GA was run using this genotypic encoding and the results showed that some improvement was achieved (typically 1-2 dBs) but the algorithm converged before any greater improvements were found. Using this encoding scheme the GA optimised the number of elements to vary within the excitation set and this approach did not result in good performance. The GA was coded to select two elements from the excitation set, but improvements to one value could be negated by the variation of the second element.

A local (Non GA) search algorithm was set up to optimise the same GA solution shown in Figure 5-3, but this time optimised a single element at a time. The algorithm chose a single element at random and replaced its existing weighting with a randomly generated new value. If there was an improvement the original pattern was replaced with the new. This simple algorithm showed immediate improvement in results and reduced the sidelobes by up to an additional 4.0dB (maximum sidelobe level -24dB) after 2,000 iterations. The combination of GA and local search required the evaluation of 62,000

trial patterns. Five bit binary encoding would give a search space containing a possible 1.0737×10^9 (for a single element) so the approach proved very efficient.

Re-tasking the same local search to operate on two elements at a time produced even better results (up to -29dB sidelobes) that were comparable with the Taylor weighting. The disadvantage was that a further 5500 iterations were required to achieve the improvement. To further improve the local search, the random replacement of excitations was replaced with a simple addition or subtraction of 0.05 to the weighting. This method refined a solution over time and achieved the same performance as the random replacement after just 1300 iterations. In addition, the more iterations that were completed, the better the solution became. After 10,000 iterations, a pattern with sidelobes 39dB down was achieved.

This result showed that use of a simple GA coupled with a simple local search algorithm could find exceptionally good excitation sets.

The original GA searched for each element's amplitude weightings individually, and this produced jagged tapers (Figure 5-3), quite unlike the bell-shaped taper of a Taylor. Generally in real array hardware, it is good practice to maintain a modest difference in magnitude between neighbouring element's weightings - large differences should be avoided. Controlling the array in this manner can help to prevent local hot spots forming on the array surface, and simplify the cooling system used. Prior work by researchers such as Rodriguez et al [113] has introduced objective measures into the fitness function to reward solutions that minimise the difference between successive excitation values. There are also numerous examples of minimisation of the difference between minimum and maximum excitation values. This practice can be quite constraining and may limit the success of the optimisation, particularly when minimising sidelobe levels where large differences are required maximum and minimum excitation values (such as a Taylor-like distribution).

In order to promote modest changes between excitation values of neighbouring elements, a new method of genotypic encoding of excitation sets has been developed. Experimental work showed that the amplitude component of the excitation set can be represented by using combinations of sine waves to represent the shape of the taper. The shape can be made up by overlaying different sine waves and sampling the resultant pattern to represent the excitation values. This new method will be hereon referred to as 'sine wave superposition' or 'SWS'.

The genotype encoding is implemented by first generating a chromosome ch_1 containing four random numbers. Using the same decoding method as described above (where the number of values greater than 0.5 are counted) the chromosome defined how many sine waves to use in the construction of the weighting. As at least one chromosome is required for the algorithm to function, the value obtained indicated how many *additional* sine waves to use in the range of zero to four.

A second chromosome ch_2 contained five random variables between zero and one, and these represented the frequency of each potential sine wave. A third and final

chromosome ch_3 also contained random variables between zero and one and these were used to represent the phase shift of the sine wave.

Each of the n sine waves were generated using:

$$y = \sin(ch_{2_n} R + ch_{3_n} \pi) \qquad \textbf{Eq. 5-3}$$

where $0 < R < \pi$,

ch_{2n} is the n^{th} sine wave frequency coefficient contained in ch_{2n},
ch_{3n} is the n^{th} sine wave phase shift coefficient contained in ch_{3n},

For example, if ch_1 once decoded, indicated that two more additional sine waves should be generated (three in total), the relevant values within ch_2 and ch_3 were used in Eq. 5-2. Once each of the four sine waves have been formed, they are overlaid and the maximum value at any value R is calculated. Figure 5-4 shows a number of overlaid sine waves and the resultant maximum values. Any values below zero were ignored.

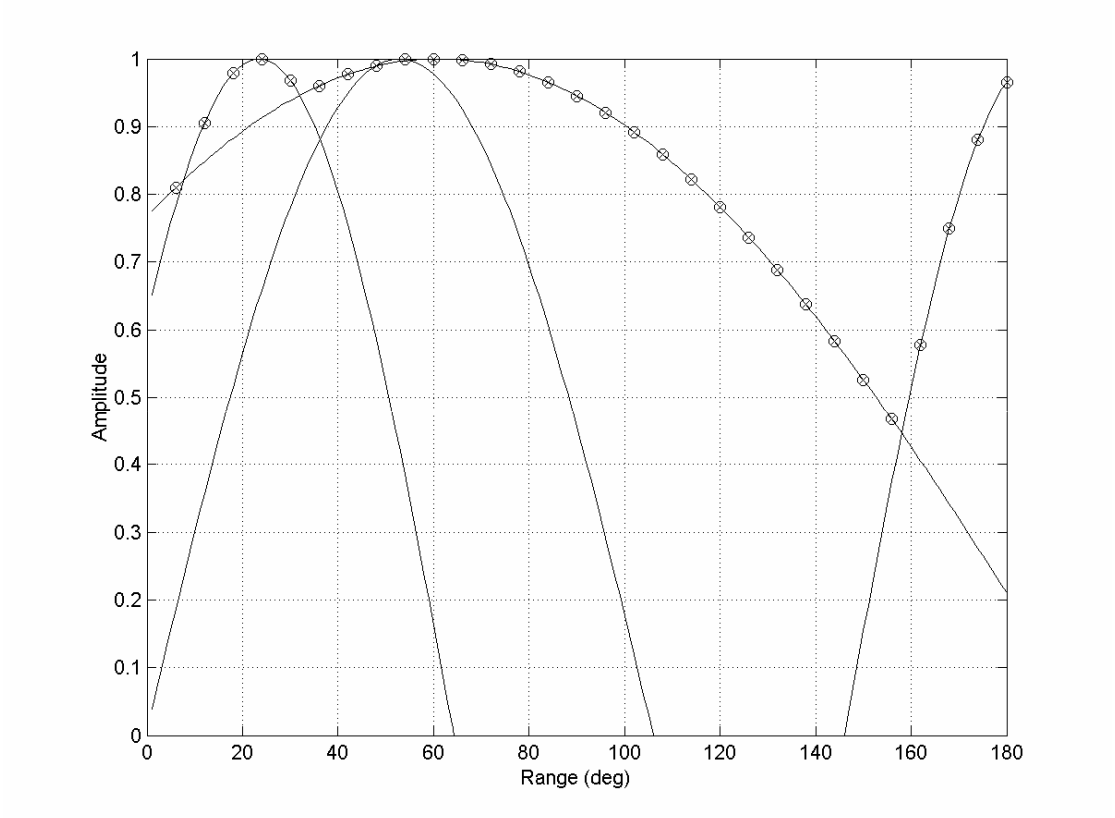


Figure 5-4 SWS Method for forming amplitude tapers.

The angular range was divided into thirty evenly spaced samples (6º each) shown by the circular markers on the plot. These values are used directly as the thirty amplitude weightings for the array.

When using the SWS encoding method, the GA is in effect searching for a combination of sine waves that result in a good performing amplitude taper, rather than searching directly in amplitude decision space. This technique has the important property of

preventing large excitation differences between neighbouring elements, ensuring a smooth taper across the taper.

The results were much more impressive and eliminated the need for a secondary local search. A simple single sine wave shaped taper, produced sidelobes 24 dB down which already bettered the best solution found by optimising each element's weights individually. Figure 5-5 shows the amplitude taper found and resultant radiation pattern (sidelobes -29dB) for one such run of a GA using this encoding method. The similarity with the optimum Taylor solution is clear.

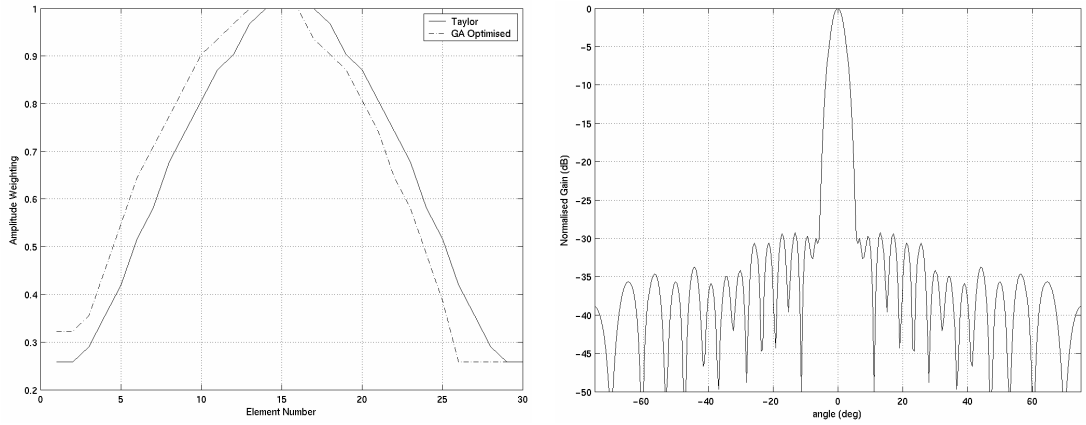


Figure 5-5 SWS Optimised Taper and Radiation Pattern

Epistasis can be described as the suppression of a gene by the effect of an unrelated (epistatic) gene. In Genetic Algorithms, if this goes unnoticed it can be a bad thing, however the above SWS encoding is designed to take advantage of epistasis.

It is fair to say that the contents of chromosome one (that contains the number of sine waves that will contribute to the pattern) is epistatic as its decoded value determines how much of the information in chromosomes two and three is used. It is capable of suppressing large parts the information contained in chromosomes two and three.

In the SWS encoding method, mutation can make both large and small jumps in the search space, dependent on which chromosome it is applied. Mutating chromosome one is likely to make a large jump as the addition of another sine wave should make a significant difference to the final excitation set. Mutation of the chromosomes that represent the frequency or phase shift variables may have a less severe effect.

It is perfectly acceptable to design custom mutation operators for each chromosome and apply them with individual probabilities of occurrence. It would be inefficient to assume that one type of mutation operator is suited to all forms of genotype.

5.3.2. Steered Arrays

It is important to determine the success of the SWS encoding on steered arrays, as fixed boresight patterns are of limited application.

The mainlobe can be steered quite simply by recalculating the phase taper. Mainlobe gain is lost as the array is steered away from boresight and the beamwidth tends to widen. A uniformly illuminated array (all amplitude values equal to one), behaves as in Figure 5-6 when steered to 60º.

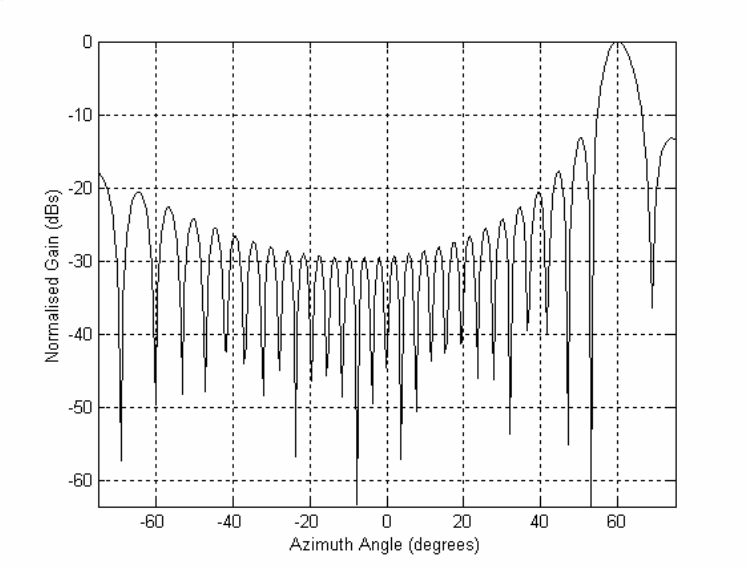


Figure 5-6 Steered Linear Array

The SWS method produced a respectable reduction in the sidelobe levels from -13.3dB to -26.1dB when compared with the uniformly illuminated pattern. The phase taper was pre-calculated and not subject to optimisation. The resultant amplitude taper and radiation pattern are shown in Figure 5-7 and Figure 5-8.

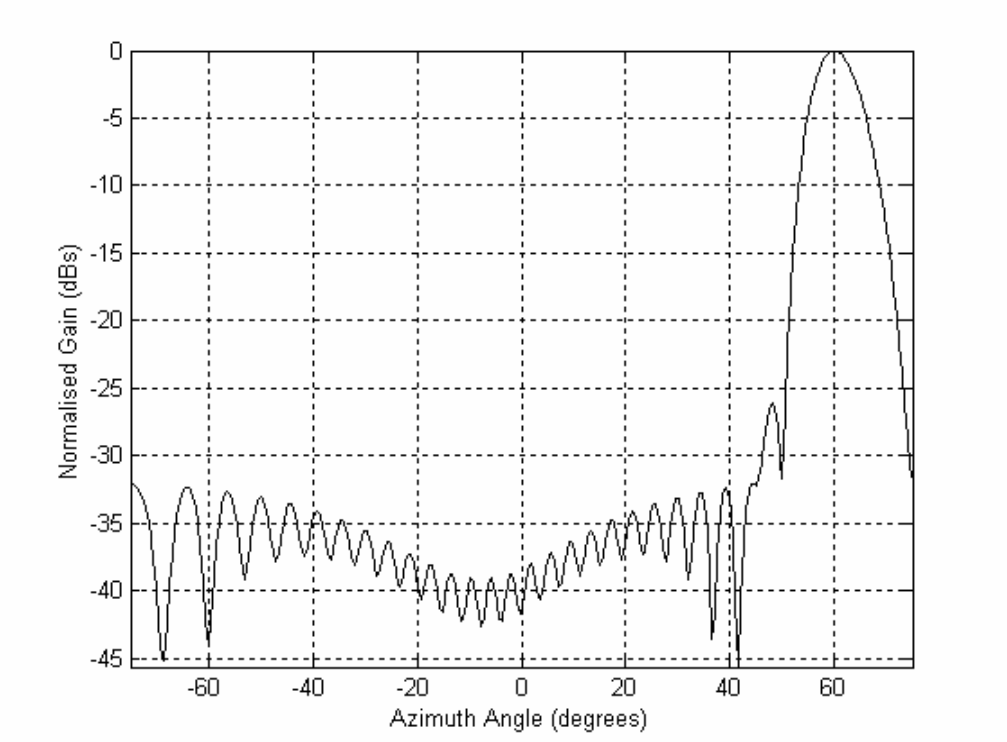


Figure 5-7 Steered Array Optimised Using SWS.

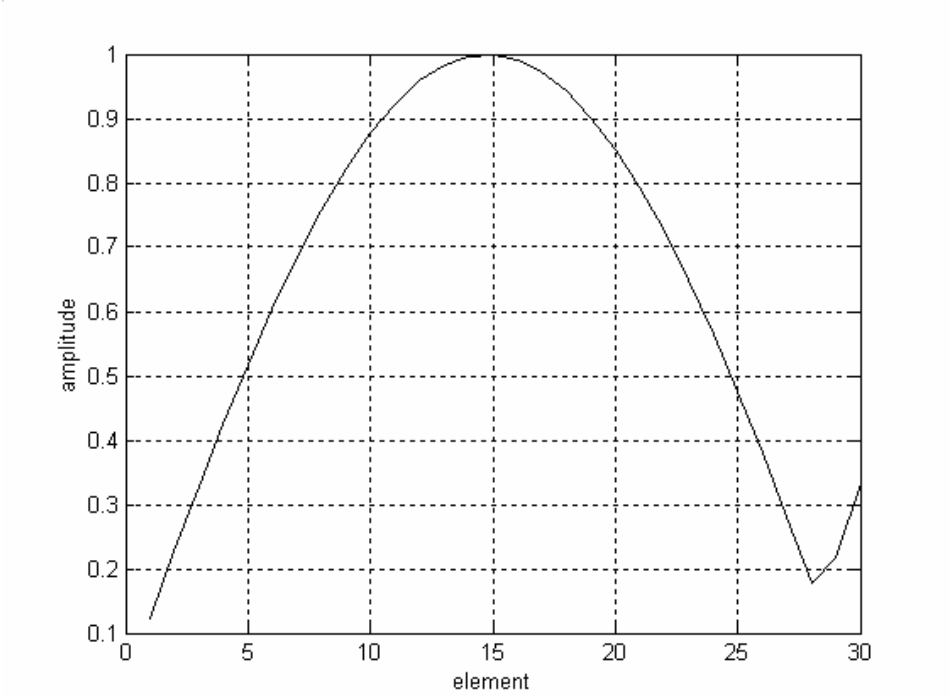


Figure 5-8 Optimised Amplitude Taper for Steered Array

5.4. Optimisation of Pattern Nulls

Classical illumination functions such as the Taylor series generate regularly spaced sidelobe positions in real array applications. In a defence environment, an antenna array can be subjected to electronic attacks, where high-powered radiation is directed at the array to effectively 'jam' its operation. One means of countering jamming is to steer or produce a null (position of minimum gain) in the array's radiation pattern focused at the direction of the incident radiation. Similarly it may also be important to reduce the intensity of unwanted returns from local structures, once the array has been installed.

It is desirable to introduce the nulls in the pattern while maintaining the sidelobe level performance so the number of objectives that describe a good pattern inevitably rises. A new GA was set up to attempt to simultaneously provide good sidelobe performance and deep nulls at two azimuth angles, chosen at random to be equal to 48.09º, and 65.03º. These values correspond to sample points where the radiation pattern was calculated.

The SWS method was used as a chromosome encoding method, and a new fitness function was formulated. The fitness F of a solution n was given by:

$$F_n = f_1 - f_2 - f_3 \qquad \text{Eq. 5-1}$$

where f_1 was the difference in magnitude between the mainlobe gain and the maximum sidelobe level, f_2 was the pattern gain at 48.09º and f_3 was the gain at 65.03º. This formulation allows F_n to be maximised. The GA was set to run for three hundred generations, and all other settings remained as in the sidelobe minimisation examples

above. The fitness function worked well and ensured that good sidelobe performance was complemented by deep nulls in the desired locations. Fitness functions such as this one highlight the importance of the relative magnitudes of each objective measure. In this example, if f_2 and f_3 are small values compared with f_1 they will have little influence in the evolutionary process and f_1 would be dominant in the results set. All three objectives are direct measures taken from the pattern, and are all gain measurements of one form, but care must be taken when combining very different types of objective measurements. An optimised radiation pattern found during this null optimisation example is shown in Figure 5-9.

The GA produced a final solution with a maximum sidelobe level of -22.62dBs and the nulls at 48.09º and 65.034º were -72.75dBi and -71.44dBi respectively.

In practice, null steering in this manner relies on accurate information of the direction of the incident jamming - small errors could result in a peak rather than a null being directed towards the jammer. It is also more difficult to finely control the pattern of small arrays.

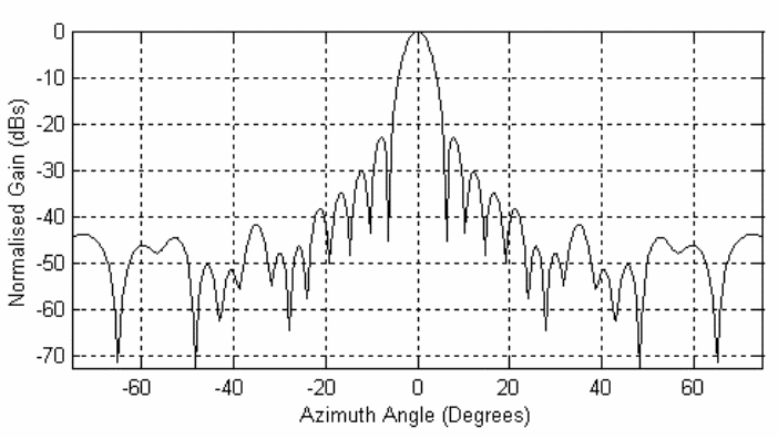


Figure 5-9 Optimised null positions and sidelobe level example.

For systems having course digital control, it may be better to evolve an excitation set that produces low magnitude sidelobes in a particular region to account for inaccuracies. This optimisation could be achieved by minimising the maximum sidelobe level in a certain angular region, with the region being centred on the estimate of jamming source incidence.

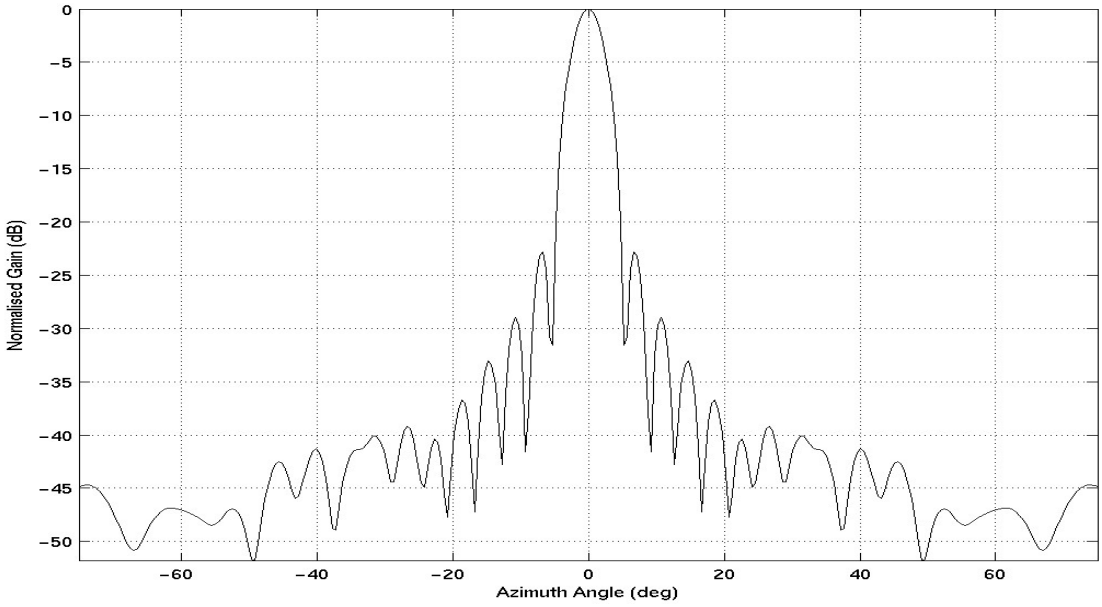


Figure 5-10 Pattern Optimised for Low Sidelobes in the 48º to 70º Region.

An example of an optimised pattern that meets this requirement is shown in Figure 5-10.

A number of different fitness functions were used during experimentation stages. For example in order to gain some measure the sidelobe levels in the 48º to 70º region, the average value can be taken and minimised in the fitness function but this resulted in poor performance. When taking an average, the benefits of a deep null can be countered by a high magnitude sidelobe. Much better performance was achieved by measuring the maximum sidelobe value in the region, and minimising that value in the fitness function. Concise objective measures are essential, as they alone guide the genetic algorithm's search towards optimal solutions.

5.4.1. Phase Only Optimisation

All the examples above concentrate on optimising the amplitude component of the excitation set. Generally, a phase taper is calculated in order to steer the mainlobe in a certain direction, so altering phase values can undesirably shift the mainlobe. For this reason, phase value optimisation is best used sparingly.

It follows that optimisation of a small number of phase values should be able to steer nulls into the pattern at desired angles. The question remains as to which phase values in the pre-calculated taper to optimise, and what their new values should be.

Following the same basic GA outline as used above, a phase-only optimisation was set up with the primary purpose of producing nulls in a Taylor weighted pattern. A Taylor weighting was calculated and applied to the same thirty-element array as used above. In addition, the pattern was steered to 35º.

With no additional phase shifts applied, the nulls on the right hand side of the mainlobe were located at angles of 42.1º, 45.6º, 50.6º, 57.1º and 65.0º (Figure 5-1).

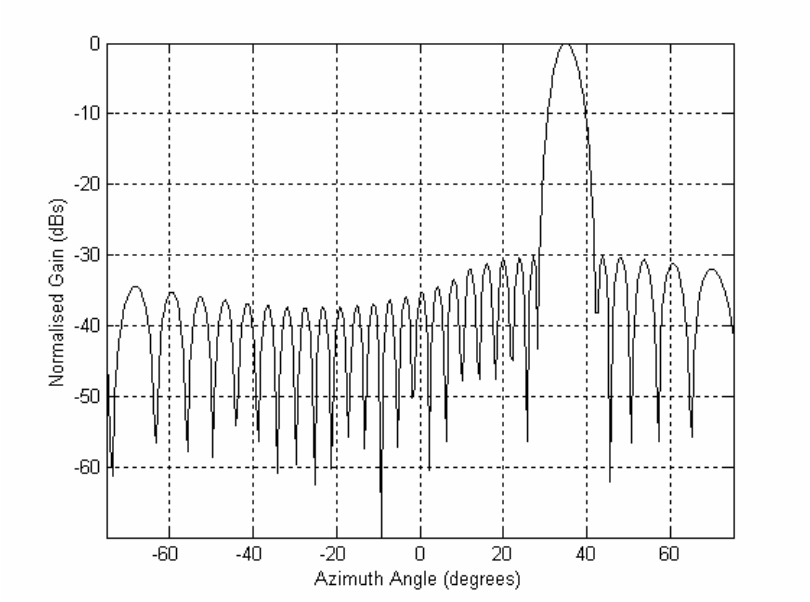


Figure 5-11 -30dB Taylor Weighted Steered Pattern

In order to test the effectiveness of the GA on this problem, two desired null positions were chosen deliberately to be away from the nulls in the original pattern. The desired nulls were set at 47.5º and 60º.

A two-chromosome GA was used with the first chromosome encoding the choice of to which elements to apply the additional phase shift, and the second chromosome encoding the amounts of additional phase shift to apply.

The fitness function was formulated to simply minimise the angular differences between two nulls closest to the desired null positions. While this process produced good results, the magnitude (depth) of the two nulls varied significantly (>30dB) so a second measure was included in the fitness function to minimise the difference in magnitude between the desired nulls. The fitness function became a simple sum of objective measures and the null depth measurements were scaled to a similar range as the angular objective measures (normalised to their maximum values) to prevent them dominating the fitness value calculation.

After two hundred generations, an acceptable pattern was found. The result showed the GA had converged on a solution that optimised elements 17 and 23 and had shifted their phases by an additional 44.4 and 14.9 degrees respectively. The pattern is shown in Figure 5-12.

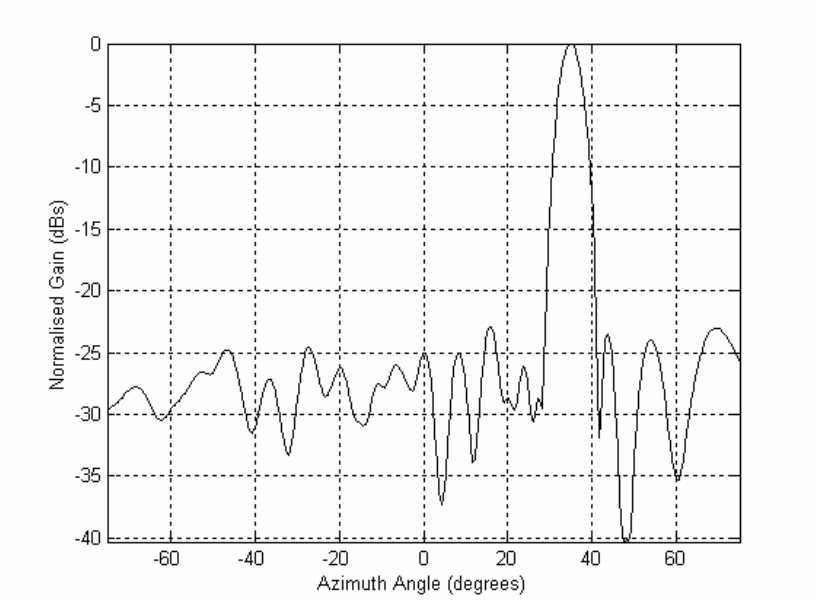


Figure 5-12 Optimised null positions in steered pattern.

The nulls were located at 47.59º and 60.05º and were of magnitudes -40.3dB and -35.3 dB respectively (5dB difference between null depths). This example showed that optimising a small number of phase values can help to move null positions and that no additional amplitude weighting is required. It should be noted that the maximum sidelobe level has increased as sidelobe minimisation was not included as an objective measure in the fitness function.

5.5. Larger Arrays

In the literature, there are few examples of the optimisation of larger arrays. Chellapilla et al used evolutionary programming to thin a two hundred-element array [114] and managed to achieve sidelobe levels of -20dB. Similarly, Fend Li and Lin Gong [115] used a GA to thin a hexagonal array, and gained results of -19.86dB. In both these examples, the number of optimisation variables increased with the array size.

The SWS encoding method has the important property of not increasing in complexity as the array size increases. The SWS method was compared with the more common way GAs are encoded (optimising the element weightings individually) on linear arrays containing one hundred elements.

The properties of a sine wave shaped taper encourages patterns to form with low sidelobes. Therefore, the SWS encoded GA started with a better solution than the best found using a GA to search for all the element weights individually. Its convergence properties were also better, as it continued to improve its best solution in later generations. The final result obtained by the SWS GA in a simple sidelobe minimisation optimisation problem produced a pattern with -38.8dB sidelobes compared with the -18.3dB optimising the elements individually.

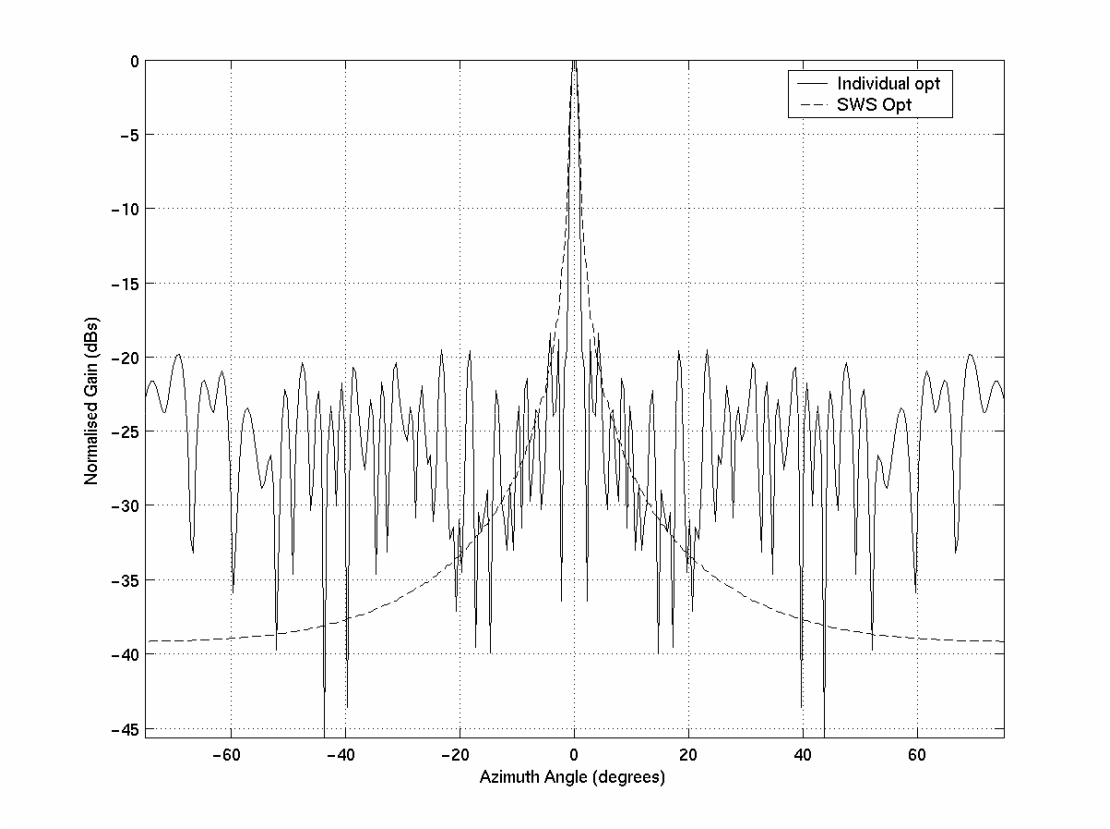


Figure 5-13 Performance Comparison on a Large Array

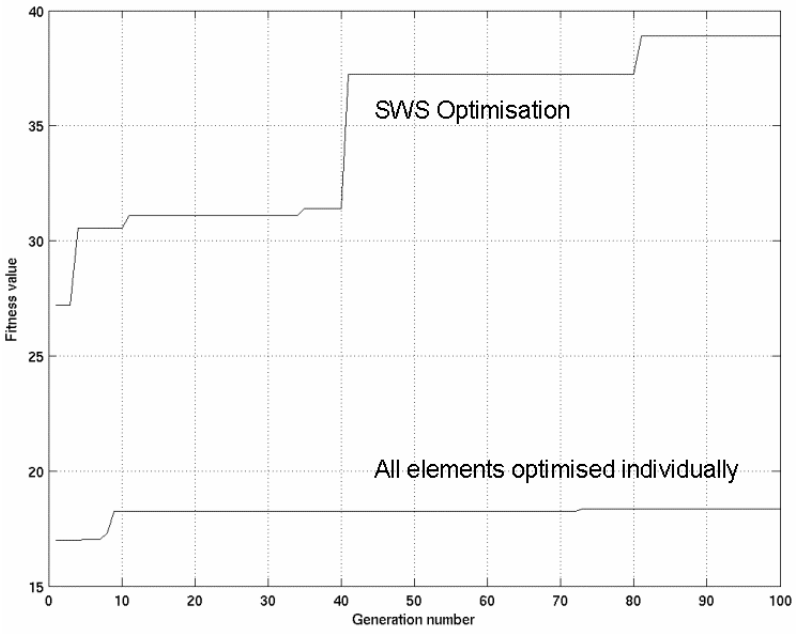


Figure 5-14 GA Convergence Comparison

It is clear from the above work, that a number of objectives are required to describe a required radiation pattern. For example, in the simple sidelobe minimisation case, it is

important to also specify a lower sidelobe level limit or a maximum mainlobe beamwidth that can be tolerated. Combining objective measures into a single fitness function is something of a trial and error process, and the manner in which the combination is performed can significantly affect results. Definition of the fitness function is difficult, as the setting of weights is non-intuitive. In practice, the combination can result in regions of search space being obscured.

While this chapter has investigated genotypic representation of optimisation variables and shows that the performance of SWS surpasses prior work, the fitness function formulation remains a limiting factor. The next chapter looks at ways of improving the evolutionary optimisation of antenna arrays by using modern 'multiple objective' evolutionary algorithms.

CHAPTER 6
Multiobjective Evolutionary Optimisation

6. Multiobjective Evolutionary Optimisation

6.1. Pareto Optimality

The Italian economist Vilfredo Pareto stated in 1896 a concept known today as the 'Pareto optimum' that constitutes the origin of research in multiobjective optimisation [116]. According to this concept, the solution to a multiobjective optimisation problem is normally not a single value, but instead a set of values called the Pareto set.

Many real-world problems involve simultaneous optimisation of several incommensurable and often competing objectives. Examples of conflicting objectives may include maximising speed and safety in a car, or keeping costs low and quality high in manufacturing. There are many such conflicts in terms of array antenna design optimisation. It was concluded in the last chapter that numerous objective measures are required to accurately and concisely describe a desired radiation pattern. Application of a simple GA that combines objectives in a single fitness function can only be expected to converge to a single solution.

Often, there is no single optimal solution to a problem, but instead a set of alternative solutions. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all objectives are considered. Solutions may be found that are good on one objective, but bad on another, forcing the designer to trade one objective for another. The set of optimal solutions is often called a trade-off surface, or more correctly, a Pareto optimal set. This chapter begins by describing how the simple genetic algorithm can be extended to search for multiple points on a Pareto surface.

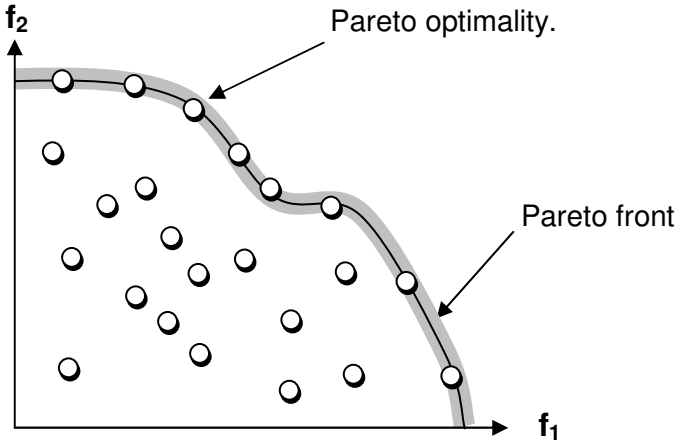


Figure 6-1 Objective Space for a Two Objective Problem, Maximising f_1 and f_2.

Figure 6-1 illustrates a Pareto front in objective space. The solutions on the grey curve define the Pareto optimal set (where all objectives are equally important). All other solutions are non-optimal. The choice of which solution to choose from the Pareto set is based on preference information at the time the decision is made. Importantly the extra information provided by the Pareto set can improve confidence in decision making because alternative solutions are known.

Multiobjective or multi-criterion optimisation problems have largely been avoided due to their complexity and hence the difficulty in solving them using classical optimisation methods. Problems typically experienced include:

1. An algorithm has to be applied many times to find multiple Pareto-optimal solutions as only one solution is found during each run (i.e. little or no diversity in the results obtained from run to run),
2. Many algorithms demand some knowledge about the expected end result,
3. Some algorithms are sensitive to the shape of the Pareto-optimal front,

6.2. Problems with Simple GAs and Weighted Sum Fitness Functions

Consider Figure 6-1. This two-objective problem has a non-convex cost function. When using a fitness function that contains a sum of objectives, weighted or otherwise, you are in fact defining a straight line in objective space on which the solution must lie. If the cost function is non-linear, the straight line may pass through a number of possible solutions on the curve as in Figure 6-2.

In weighted sums, the gradient of the line is defined by the weights that are used. If the total fitness of a solution is given by F where

$$F = w_1.f_1 + w_2.f_2 , \qquad \textbf{Eq. 6-1}$$

the gradient of the line is given by,

$$\text{Gradient} = \frac{-w2}{w1} \quad . \qquad \textbf{Eq. 6-2}$$

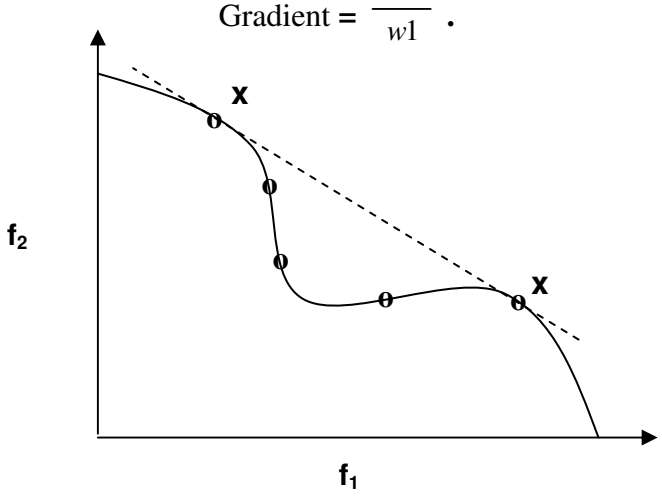


Figure 6-2 Non-Convex Cost Functions

In problems with cost functions such as in Figure 6-2, it is not unusual for the GA to oscillate between good solutions either side of concave regions of the curve. The two solutions marked 'x' on the Figure indicate possible oscillation points. Oscillations of this type are reported in literature on the applications of GAs, but are not always correctly attributed as a problem with sums (or weighted sums) of objectives [117].

6.3. Finding the Pareto Front

Let us consider an antenna optimisation problem with three objectives – minimisation of sidelobe level (F1) and beamwidth (F2), and maximisation of transmit power (F3). If we generate 100 random excitation sets and apply them to the antenna, we can produce a plot in objective space that shows the success of each set in terms of the objectives. We would typically see a distributed sample of solutions as in Figure 6-3. In the Figure, the three objectives (F_1, F_2 and F_3) have values between zero and one and have been encoded for maximisation.

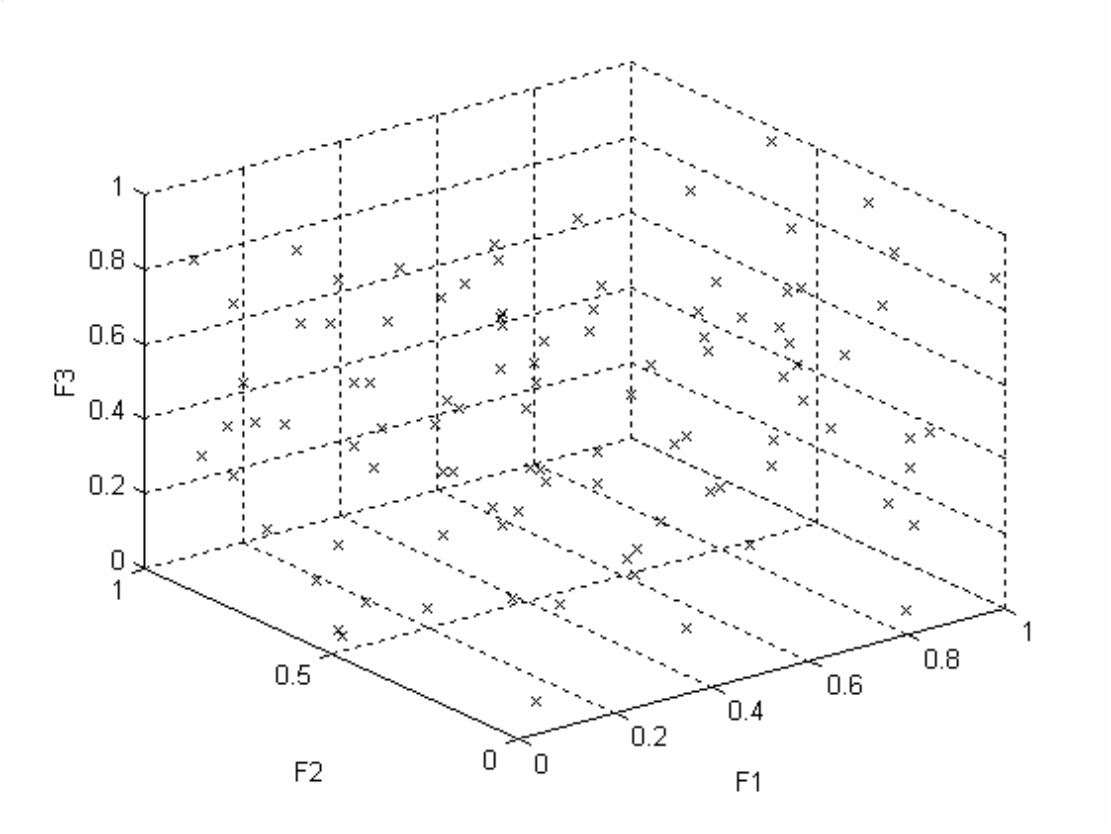


Figure 6-3 Initial Sampling of Objective Space

At this stage we do not know if we have found any optimal solutions, nor can we can make any conclusions regarding the shape of the Pareto surface.

Evolutionary multiobjective algorithms are concerned with finding or approximating the Pareto optimal solutions by iterating a random (or seeded) set of initial solutions and allowing environmental selection mechanisms (as used in the GA) to guide the solutions towards their optimal values.

6.4. Issues in Evolutionary Multiobjective Optimisation.

Before some evolutionary algorithms are described in more detail, it is necessary to introduce some important issues and terminology in multiobjective optimisation.

The first such issue is one of solution density - how to maintain a diverse Pareto set approximation. If all solutions found occupy a small but densely packed region of objective space, then few useful trade-offs can be made between them. A good evolutionary optimisation algorithm will contain mechanisms to ensure diversity in the solutions found. It is of course possible that the objective space is very densely packed and narrow, but without encouraging diversity we cannot tell if it is *meant* to be that way.

One method for encouraging diversity is known as fitness sharing. Similar solutions can have their fitness values shared if they are within a certain Euclidean distance (in

objective space) of each other. The fitness values are reduced by dividing them by the number of local solutions. The fitness reduction reduces the selection pressure on these solutions and so favours solutions that are distant from others in the population (they will not have had their fitness values reduced by sharing).

The second issue common to all iterative algorithms is how to ensure good solutions are not lost. A good optimisation algorithm stores good solutions, a better one uses them to find still better solutions.

The third and most important issue is how the solutions are guided towards Pareto optimality (convergence). The convergence mechanism is perhaps where different evolutionary algorithms differ the most and also what makes some more successful than others.

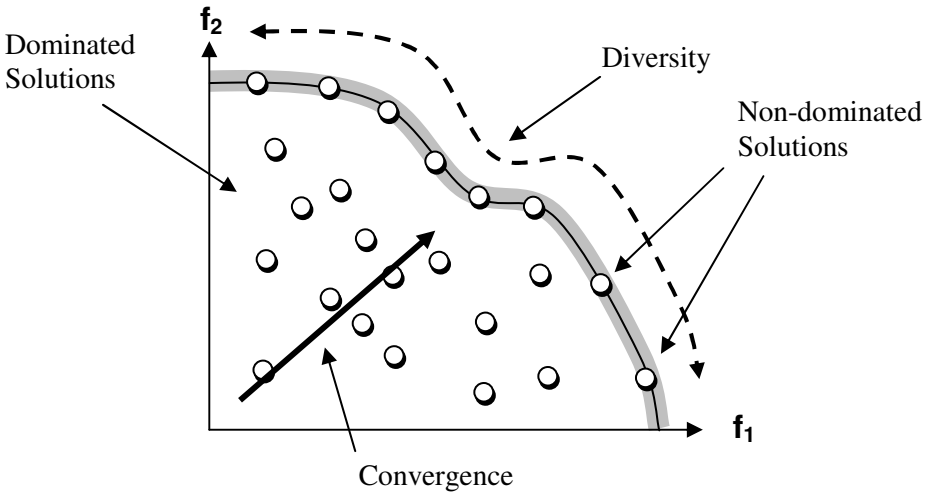
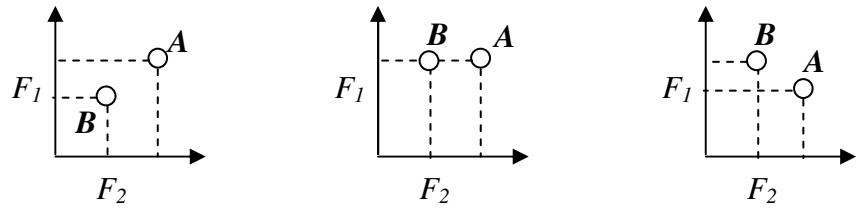


Figure 6-4 Multiple Objective Optimisation Terminology

A very important concept in multiobjective optimisation is one of dominated and non-dominated solutions.

Dominated solutions are non-optimal and are dominated simply because better solutions exist. Non-dominated solutions are the current best Pareto set approximation. They are not necessarily the global optimums.



A dominates ***B.*** ***A*** has higher values of both F_1 and F_2 than ***B***.

A dominates ***B.*** ***A*** has higher value of F_2 and equal value of F_1.

Neither A nor B are dominated. Both solutions are non-dominated.

Figure 6-5 Dominated and non-dominated solutions (maximising F_1 and F_2).

When maximising objectives, a solution ***A*** is said to dominate solution ***B*** if its objective values are at least equal to those of ***B*** and better (higher) in at least one objective (when minimising a least one must be lower). A solution is described as non-dominated if no other solutions dominate it.

6.5. A Survey of Evolutionary Multiobjective Optimisation Algorithms

This section describes some methods of evolutionary optimisation that use measures of dominance in different ways in order to guide the solutions towards the Pareto front. The algorithms all have genetic algorithm 'core' i.e. familiar crossover and mutation operators but differ significantly in their selection, mating and diversity methods. Some of the algorithms are now quite old and little used today, but are included as they demonstrate how the state of the art has advanced as understanding of the issues in multiobjective optimisation have improved.

6.5.1. Aggregation Based (1980's)

Aggregation based methods are ubiquitous in genetic algorithms and typically use a weighted sum of objectives usually with a standard genetic algorithm [61,62,63,65].

The method can create a single solution that is strongly non-dominated. The main problem that often goes unnoticed is that it does not generate Pareto optimal solutions in the presence of non-convex search spaces, regardless of the weights applied to each objective.

6.5.2. Criterion-based: Schaffer's Vector Evaluated Genetic Algorithm (VEGA) (1985)

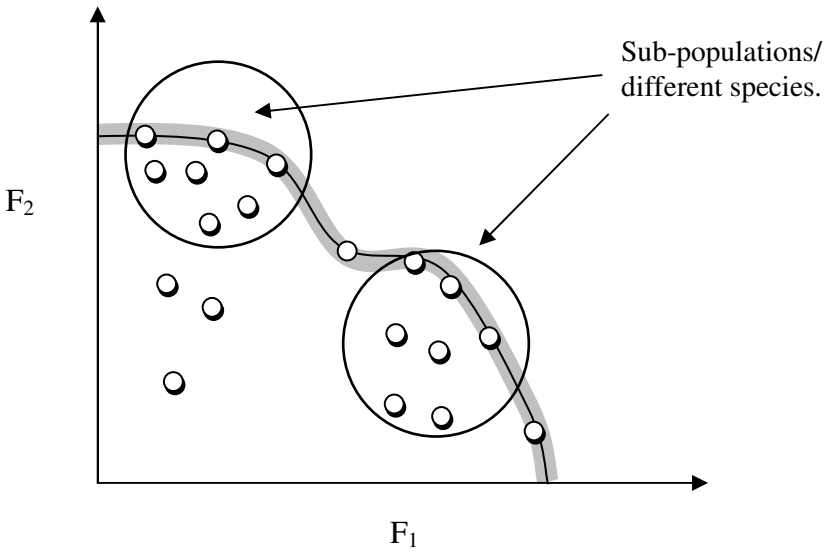


Figure 6-6 Effect of sub-populations on convergence.

Schaffer [118] modified the simple genetic algorithm by performing independent selection cycles according to each objective. The selection method is repeated for each

individual objective to fill-up a portion of the mating pool. Then the entire population is thoroughly shuffled to apply crossover and mutation operators. The shuffling is performed to achieve the mating of individuals of different sub-population groups. Sub-populations are locally non-dominated but may be globally dominated [119]. The algorithm worked efficiently for some generations but in some cases suffered from its bias towards some individuals or regions (mostly individual objective champions). The number of sub-populations also limits diversity, as solutions tend to cluster together.

6.5.3. Fonseca/Fleming's Multiobjective Genetic Algorithm (MOGA) (1993)

Fonseca and Fleming [120] proposed an algorithm in which the rank of a certain individual corresponds to the number of solutions in the current population by which it is dominated.

Fitness assignment is performed by (1) sorting the population according to rank, (2) assigning fitness values to individuals by interpolating from the best rank to the worst according to some function, usually linear, but not necessarily, (3) averaging the fitnesses of individuals with the same rank so that all of them will be sampled at the same rate. This procedure keeps the global population fitness constant while maintaining appropriate selective pressure, as defined by the function used. Fitness sharing was used to encourage diversity in the Pareto optimal set. The algorithm can result in premature convergence due to a large selection pressure placed on individuals. Performance is dependent on an appropriate sharing factor.

6.5.4. Srinivas and Deb's Non Dominated Sorting Genetic Algorithm (NSGA) (1994)

In this algorithm, a 'dummy fitness' (a function of a solution's convergence and diversity measures) is assigned proportional to the population size. The solutions are then sorted according to a non-dominated sorting algorithm [67]. Then the group of classified individuals is ignored and another layer of non-dominated individuals is considered. The process continues until all individuals in the population are classified (Non-dominated sorting). This ranking scheme can be seen in Figure 6-7.

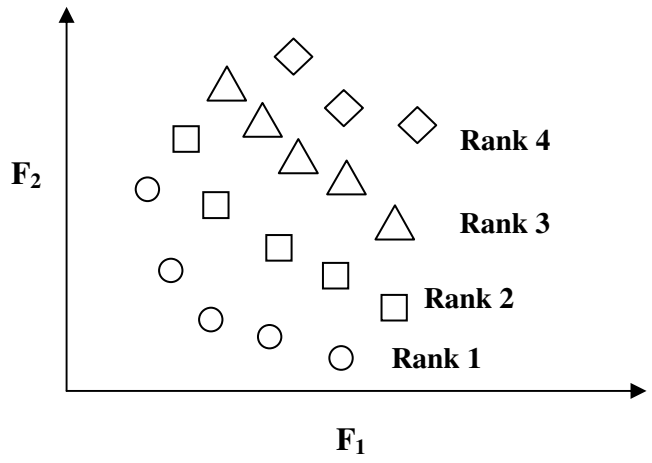


Figure 6-7 An example of non-dominated sorting (minimising both F_1 and F_2).

A stochastic remainder proportionate selection (Section 4.3.1) was used for this approach. Since individuals in the first front are assigned the maximum fitness value, they always get selected more often than the rest of the population. Fitness sharing is used to encourage diversity.

The algorithm can handle any number of objectives and implements sharing in the parameter value space instead of the objective value space. The algorithm is less efficient computationally compared with other methods and is sensitive to the value of share factor used. An update to NSGA, named NSGA II has been released [121]. NSGA II, improves upon NSGA by incorporating elitism, to ensure good hard to find solutions are not lost. Also the dependence on the sharing parameter is removed. Shortly after NSGA II's release, a modified version was published that controls the elitism operator to ensure that a set number of front members exist in the population at all times [122]. NSGA II was judged as the fast-breaking paper in engineering by Web of Science (ESI) [123] in February 2004.

6.5.5. Horn and Naffpliotis' Niched Pareto Genetic Algorithm (NPGA) (1993)

Horn and Nafpliotis [124] proposed a tournament selection scheme based on Pareto dominance. Instead of limiting the comparison to two individuals, a number of other individuals in the population were used to help determine dominance (typically around 10). When both competitors were either dominated or non-dominated (i.e., there was a tie), the result of the tournament was decided through fitness sharing [125]. Population sizes considerably larger than usual with other approaches were used so that the noise of the selection method could be tolerated by the emerging niches in the population.

This approach does not apply Pareto selection to the entire population, but only to a segment of it at each run. Its main strengths are that it is very fast and produces good Pareto surfaces. The algorithm does require careful choice of the share factor and slightly larger population size to perform well. A number of trial runs may be needed to scale these parameters appropriately.

Consider Figure 6-8, two competing individuals labelled ***A*** and ***B*** and a comparison set labelled ***B***, ***C*** and ***E*** are chosen at random from the population. The competing individuals ***A*** and ***B*** are fighting for the right to reproduce (i.e. become a parent). The size of the comparison set is given by the parameter t_{dom} (equal to three in this case). In this example, both the candidates ***A*** and ***B*** dominate two out of the three members in the comparison set. '***A***' dominates solutions ***C*** and ***D*** and '***B***' dominates solutions ***D*** and ***E***.

At this point, if one of the solutions proves to be less dominant than the other, it would lose the contest and the other would be selected as a parent. In this example, both the solutions ***A*** and ***B*** are non-dominated and the contest must be decided by other means.

If both individuals are non-dominated (or equally dominated), the result of the tournament is decided by sharing: the individual that has the least individuals in its niche (defined by δ_{share}) is selected for reproduction. In its simplest form, this method equates to drawing a circle around a particular solution and counting the number of local solutions within the circle. The main problems with the algorithm is that it is very

sensitive to the parameters t_{dom} and δ_{share} and may take several trial runs to effectively set these values.

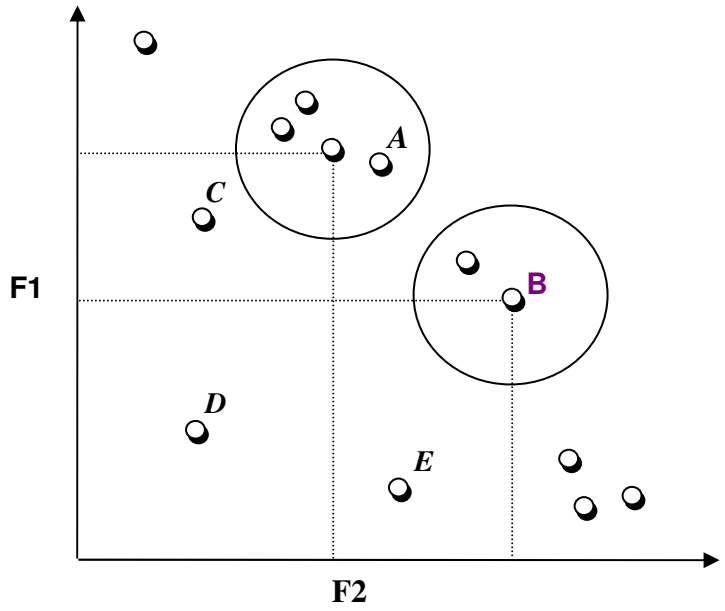


Figure 6-8 Niched Pareto Tournament Selection

6.5.6. Knowles and Corne's Pareto Archived Evolution Strategy (PAES) (1999)

Knowles and Corne's Pareto Archived Evolution Strategy [126] is a simple optimisation strategy that uses a population size of one. Uses a reference archive of previously found solutions to identify the approximate dominance ranking of the current individual.

The algorithm converges faster than most other algorithms and usually performs well but as it operates on a single solution, it cannot approximate a Pareto optimal set (i.e. it finds a single point on the Pareto surface).

6.5.7. Dominance Based: Strength Pareto Evolutionary Algorithm (1999).

The Strength Pareto Evolutionary Algorithm (SPEA) [127] is proving to be a powerful optimisation technique. A flow chart summarising the algorithm is presented in Figure 6-9. First, the external Pareto set is updated and all non-dominated individuals in the population are copied to the Pareto set.

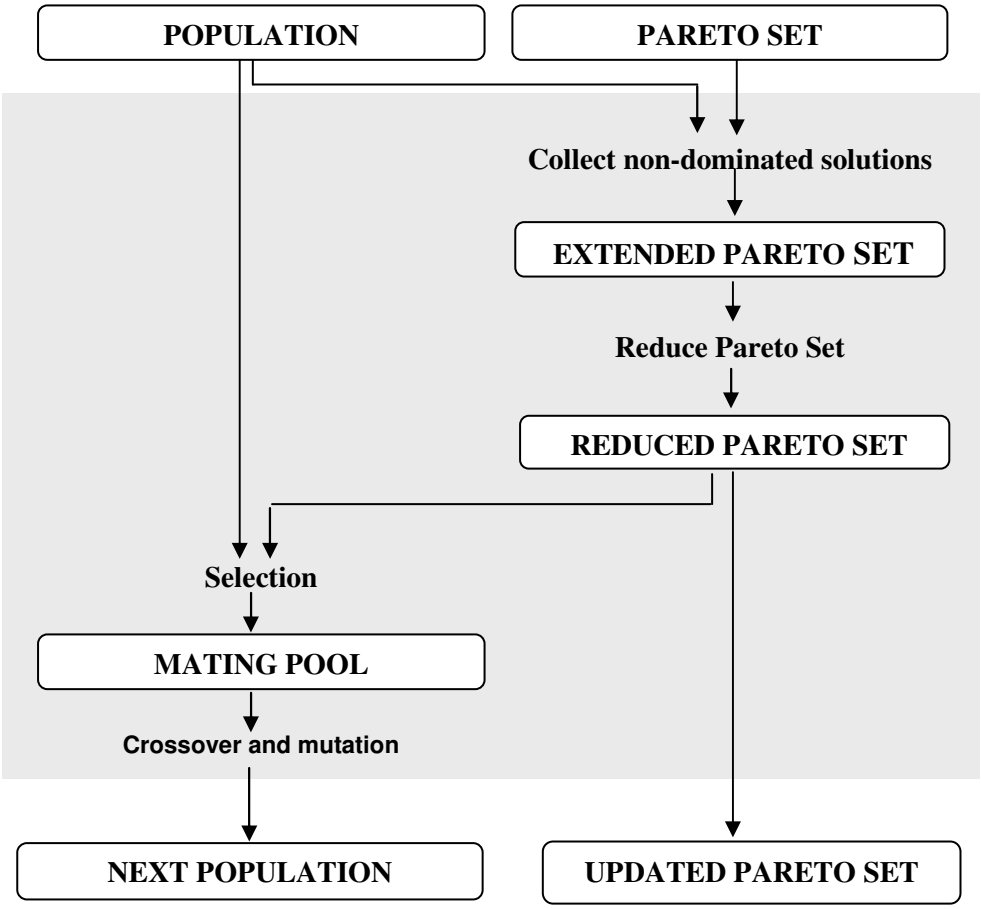


Figure 6-9 SPEA Algorithm Outline

If the number of externally stored Pareto solutions exceeds a given maximum, a reduced representation is computed by clustering. This removes solutions that lie close to each other on the Pareto front and hence encourages diversity along the front. After fitness assignment, individuals randomly picked out of the union of population and Pareto set, hold binary tournaments in order to fill the mating pool. Finally crossover and mutation are applied to the population as usual.

SPEA stores the Pareto-optimal solutions externally. At each point in time, the external Pareto set contains the non-dominated solutions of the search space sampled so far. This method ensures that Pareto optimal solutions cannot be lost, yet the population size does not restrict the number of Pareto-optimal solutions produced. Further more, the external Pareto set is used to evaluate the individuals in the population.

Since our goal is to find new non-dominated solutions, individuals are evaluated according to the number of solutions that dominate them. An example of this raw fitness assignment is shown in Figure 6-10.

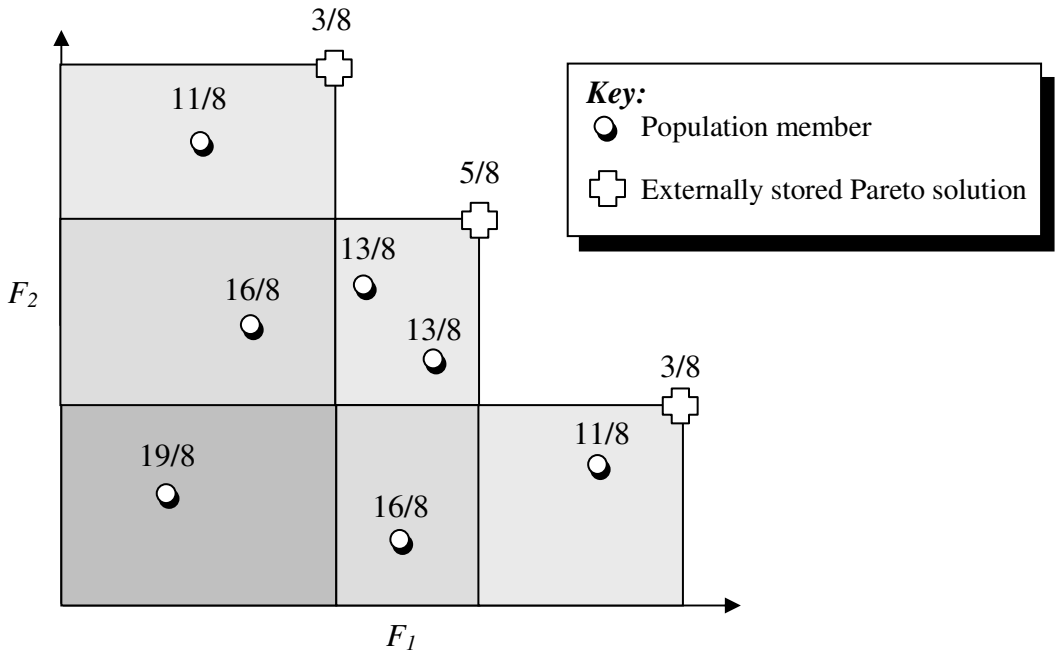


Figure 6-10 SPEA ranking scheme.

In the figure, there are three members in the external set (3/8, 5/8 and 3/8). The external set stored non-dominated solutions. The number of solutions in the population that they dominate assigns them their 'strength' values. So the top point 3/8 dominates three solutions out of the population size of seven (we add one to the denominator to ensure all non-dominated solutions have strengths less than one).

The internal population members (those marked with white dots) are assigned their strength values according to the sum of the external points that dominate them. We then add one (8/8 in this case) to the strength figure to ensure they are degraded in fitness compared with the non-dominated solutions.

Most of the evolutionary multiobjective algorithms use some kind of function to control the number of externally stored non-dominated solutions. There would be little benefit from the decision-maker's point of view, in storing all of the non-dominated solutions found. Maintaining all solutions would reduce selection pressure and slow down the optimiser.

With niche sharing methods, as in NPGA, the diversity of the Pareto front depends on the granularity of the niches. It is desirable to make them equal in size to uniformly distribute the individuals. If the uniformity does not occur, as is often the case, the fitness assignment method is eventually biased towards certain regions of the search space, and the Pareto front forms in distinct clusters according to the size of the δ_{share} parameter.

SPEA with its use of clustering, removes the dependence on a niche count, and instead initially treats each Pareto solution as an individual cluster. The clusters are then combined with their nearest neighbour in solution space, until a certain predefined cluster size is reached. Two cluster sets are selected for reduction calculating the distance of all clusters to their nearest neighbouring cluster, calculated as the average distance between pairs of individuals across the two clusters. The two nearest clusters are chosen.

The set of Pareto solutions is reduced by selecting from each of the two clusters, a representative individual and removing the other cluster members. The centroid of the cluster is selected to remain, as it has minimum distance to all other solutions in the cluster. This technique represents an important strength over other evolutionary multiobjective algorithms.

Figure 6-11 demonstrates the clustering concept. Figure 6-11(a) shows the externally stored set and the current Pareto front estimation, and (b) shows the solutions divided into clusters and the calculated centroid of each cluster (solid circle). Figure 6-11 (c) shows the reduced set.

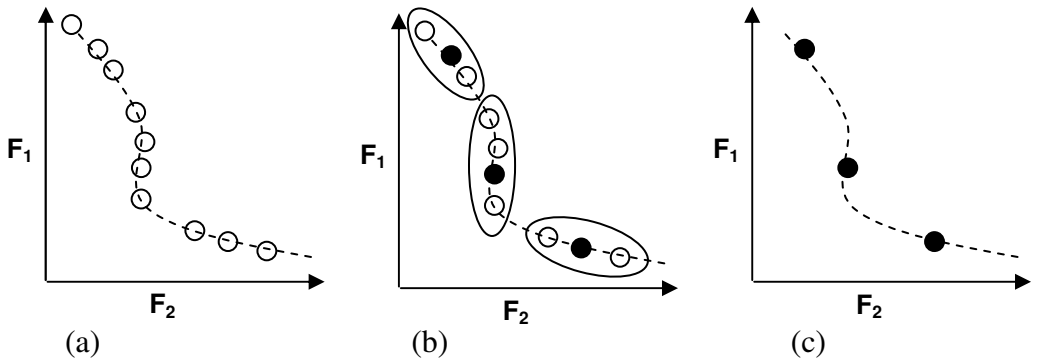


Figure 6-11 SPEA Clustering Technique.

6.5.8. Strength Pareto Evolutionary Algorithm 2, SPEA2 (2001)

Since SPEA was published in 1999, Zitler et al updated the algorithm to produce SPEA2 [128].

The main differences between SPEA and SPEA2 are that the external archive of non-dominated solutions is of fixed size in SPEA2, and the clustering technique has been replaced with a truncation function that includes a measure of local solution density (similar to a niche count, but performed on all solutions in the current population). The truncation operator helps to maintain those solutions that have proved hard to find. For detailed description of the truncation operator, please refer to [126].

The overall algorithm (reproduced from [123]) is as follows:

Input:	N	(population size)
	$\overline{N}$	(archive size) referred to as the external population in SPEA1
	T	(maximum number of generations)
Output:	**A**	(non-dominated set)

Step 1: ***Initialisation:*** *Generate an initial population* P_0 *and create the empty archive (external set)* $\overline{P_0} = 0$. *Set* $t = 0$.

Step 2: ***Fitness Assignment:*** *Calculate fitness of individuals in* P_t *and* $\overline{P_t}$.

Step 3: ***Environmental Selection:*** *Copy all non-dominated individuals in* P_t *and* $\overline{P_t}$ *to* $\overline{P_{t+1}}$. *If size of* $\overline{P_{t+1}}$ *exceeds* $\overline{N}$ *then reduce* $\overline{P_{t+1}}$ *by means of the truncation operator, otherwise if size of* $\overline{P_{t+1}}$ *is less than* $\overline{N}$ *then fill* $\overline{P_{t+1}}$ *with dominated individuals from* P_t *and* $\overline{P_t}$.

Step 4: ***Termination:*** *If t >= T or another stopping criteria is satisfied then set* ***A*** *to the set of decision vectors represented by the non-dominated individuals in* $\overline{P_{t+1}}$ *in order to fill the mating pool.*

Step 5: ***Mating selection:*** *Perform binary tournament selection with replacement on* $\overline{P_{t+1}}$ *in order to fill the mating pool*

Step 6: ***Variation:*** *Apply recombination and mutation operators to the mating pool and set* P_{t+1} *to the resulting population. Increment generation count (t = t+1) and go to Step 2.*

6.5.9. Steady state ε-MOEA

The steady state ε-MOEA developed by Deb, Mohan and Mishra [129] is an elitist approach in which the multiobjective search space is divided up into a number of hyperboxes.

Diversity is maintained by ensuring that each hyper-box can only contain one solution. The hyperboxes prevent solutions being found that are very similar to each other in one or more of the objective measures.

As with the other algorithms described above, ε-MOEA is initialised by creating and evaluating a random population of individuals *P(0) (please note the notation change from* P_t *to P(t) to represent the population. The notation change is to maintain consistency with the original publication of the algorithms in the literature)*

The resolution of each objective is chosen in order to promote the diversity. For example in the case of mainlobe gain, we may only be interested in solutions that differ by at least 1dB.

The next step in the algorithm is find the non-dominated solutions in *P(0)* and add them to an archive population, *E(0)*. The non-dominated solutions are then indexed according to which hyper-box they belong. In the rare event that two or more non-dominated solutions try to occupy the same hyper-box, their distances to the highest valued hyper-box vertice (known as the B-vector) are compared, and the solution with the shortest distance is selected.

The next stage in the algorithm works on a single solution at a time, unlike the other EAs described above. Binary tournament selection is used to choose a parent solution from *P(t)* but the selection process is modified to use a dominance check to choose between the competing solutions. In the event of both being non-dominated, one solution is chosen at random.

A second parent is then chosen purely at random from the current archive *E(t)*. Recombination and mutation operators are applied and the new offspring solution is then compared with the other solutions in *E(t)* for dominance. In the event that the offspring dominates any other archive solution occupying a hyperbox, it replaces it. If both are non-dominated the B-vector test (as described above) is used to decide the outcome. If the offspring is non-dominated with respect to the archive solutions and its hyper-box is empty, it becomes the occupier and a new member of the archive *E(t)*.

Regardless of whether or not the offspring is added to *E(t)* it is also considered as a possible replacement for a solution in *P(t)*. If the offspring dominates one or more of the population members, it replaces one of them (chosen at random). If the offspring is non-dominated with any of the population solutions, it replaces a solution at random. If the solution fails to dominate any population members it is simply rejected and 'dies off'. This strategy results in two separate co-evolving populations during the algorithm run. The advantage of bounding and dividing objective space into hyperboxes is that only solutions of interest are reported, and a degree of diversity is forced between solutions.

6.6. Comparison of MOEAs

Designers of evolutionary algorithms often quote and use a number of common standard test functions in order to demonstrate their algorithm's success in finding or approximating solutions on the Pareto front. Commonly used single-objective test functions include those produced by De-Jong [130], Schwefel [131] or Langerman [132].

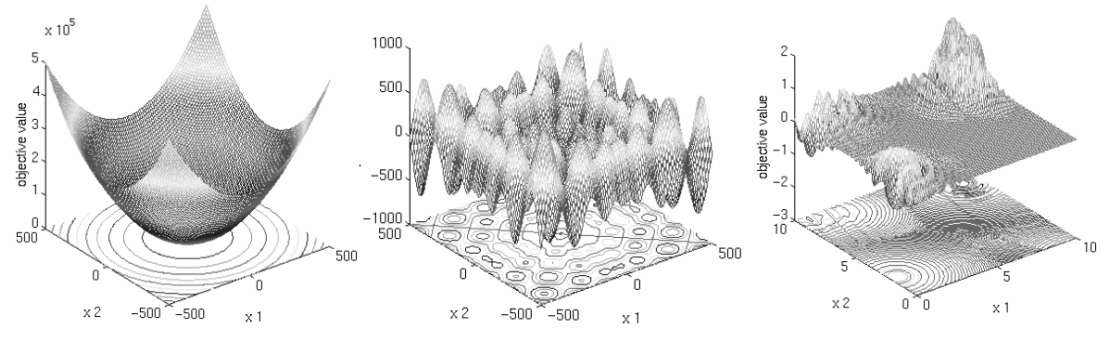


Figure 6-12 Common Test Functions

(a) De Jong Function 1, (b) Schwefel's Function (c) Langerman's Function.

The simplest of the three test functions shown above is 'De Jong's Function 1' (Figure 6-12 (a)). It is continuous, convex and unimodal function that does not present much of a challenge for a well performing optimisation algorithm. 'Schwefel's Function 2' is more of a challenge and is deceptive in that the global minimum is geometrically distant from the next best local minima (Figure 6-12(b)). Search algorithms are potentially prone to convergence in the wrong direction on this function.

The Langerman 'Test Function 3' is multimodal (Figure 6-12(c)). The local minima are unevenly distributed in objective space.

These functions are usually used for testing simple genetic algorithms rather than multiple objective variants, but one could envisage a scenario where several of them are used together to form a multiple objective problem. A number of specific two and three objective test functions are given in the literature [133, 134, 135].

While these test functions are often used to provide good generic indications of an optimiser's performance, the characteristics of antenna problem Pareto surfaces are not known. In order to investigate and compare various EA's performance on antenna optimisation problems a new representative test function is needed. A new test function specifically for assessing MOEA performance on antenna problems is described below.

6.7. Multiobjective Antenna Optimisation Test Functions.

The size of the solution space in antenna optimisation has been shown to be vast, particularly as the array size increases. The size makes exhaustive searching for optimised excitations of large arrays impractical.

If a very small array is analysed, and the degree of digital control is constrained to say 3-bit binary quantisation, it becomes possible to complete an exhaustive search within a reasonable time. The results of the exhaustive search can be used to derive the true Pareto set for the array. The results of the search effectively produce new and relevant test functions that can be used to evaluate optimiser performance.

A seven-element array with three-bit amplitude control generates a search space containing 2^{21} excitation sets (2097152 solutions). The exhaustive search showed that

the radiation patterns (azimuth cuts) produced by the array fall into four categories, namely those that produce a mainlobe with:

(1) no sidelobes, such as the example in Figure 6-13a;
(2) one sidelobe either side, Figure 6-13b;
(3) two sidelobes either side, Figure 6-13c;
(4) three sidelobes either side, Figure 6-13d.

These categories are true when the angular sampling is constrained to ±75° in azimuth. The angular sampling was set at ±75° in azimuth to reduce total computation time. Modern multifunction use of active array antennas will demand knowledge of quite different Pareto optimal operating points. For example, it may be necessary to reduce gain levels in order to lower sidelobe performance when operating covertly, and then switch to a very broad beamwidth mainlobe for a 'broadcast' transmission.

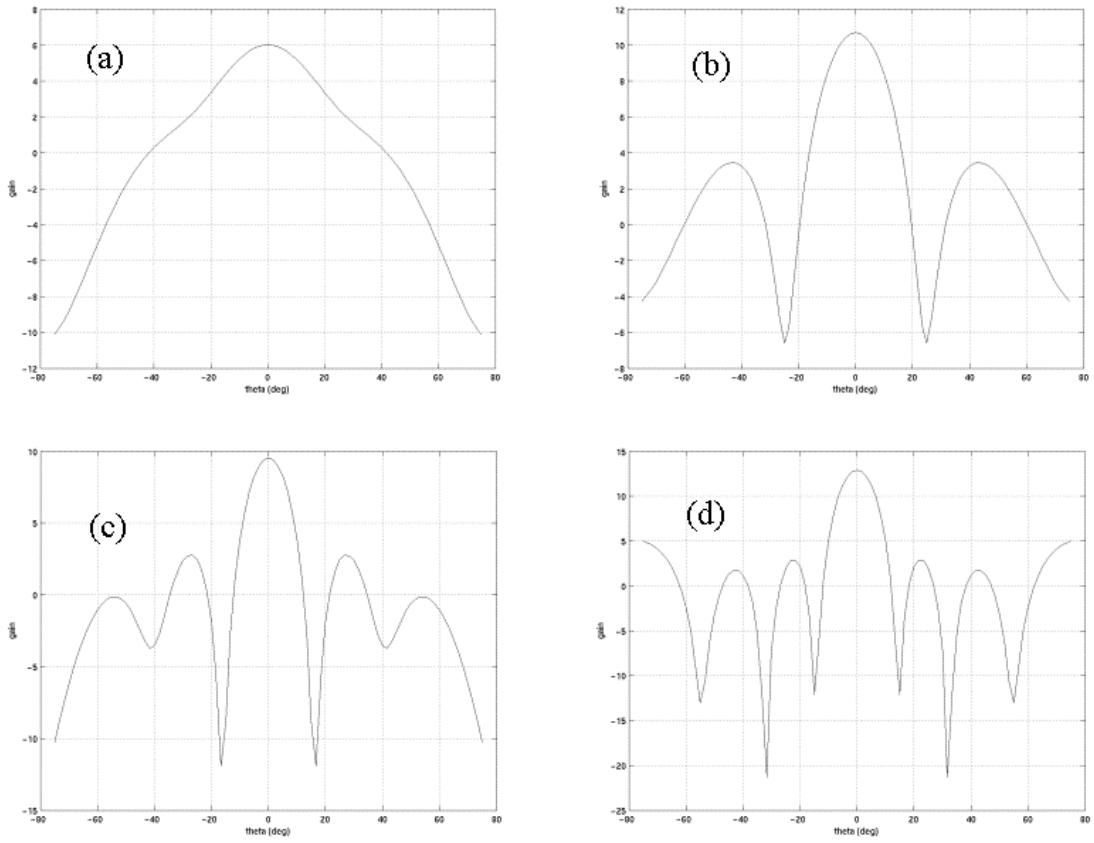


Figure 6-13 Radiation Pattern Types in the Exhaustive Search

Each of the 2^{21} excitation sets were calculated along with their resultant radiation patterns. These radiation patterns were then subject to a number of objective measures:

- Objective f_1 measured mainlobe gain,
- Objective f_2 measured the magnitude of the difference between the mainlobe gain and the maximum sidelobe level in the pattern,
- Objective f_3 measured the magnitude of the difference between the mainlobe gain and the first sidelobe (closest to the mainlobe) in the pattern.

Given the complete set of data, non-dominated sorts of the data set were completed to determine the true Pareto optimal solutions when objectives f_1 and f_2 were considered (a two-objective problem) and when objectives f_1, f_2 and f_3 (a three-objective problem) were considered.

6.7.1. Two Objective Pareto Set

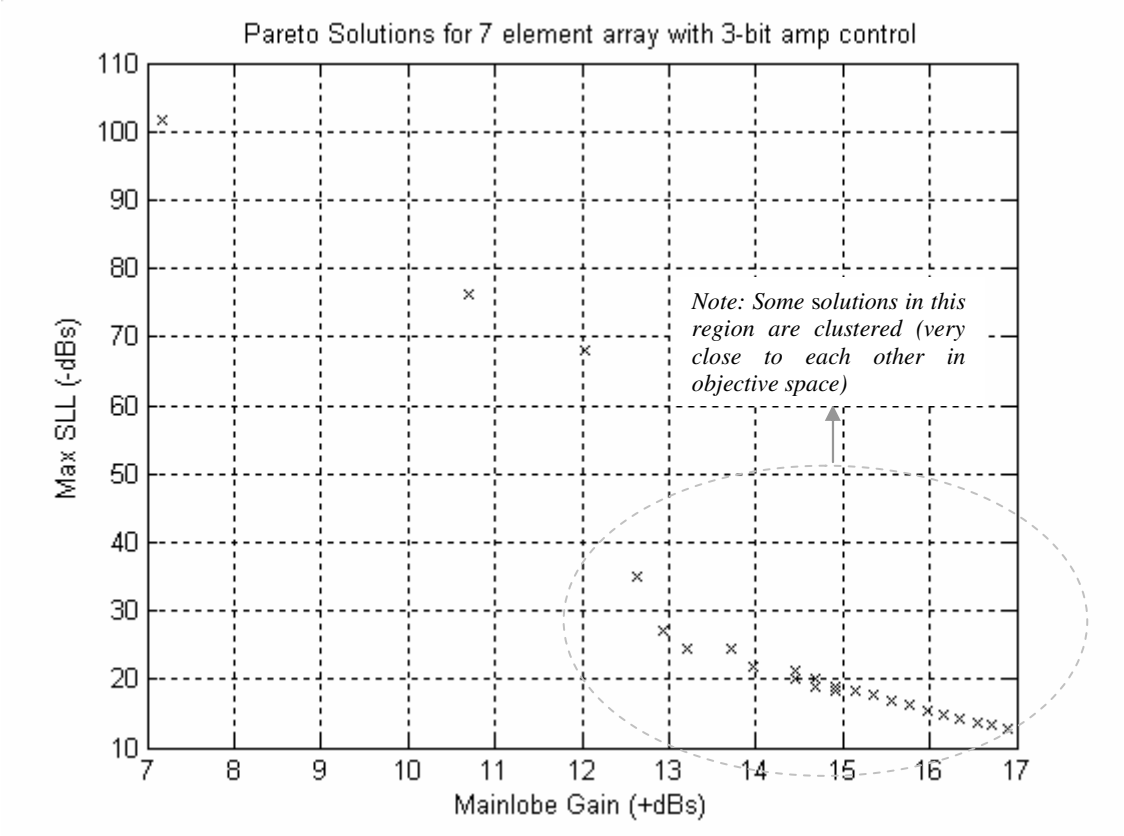


Figure 6-14 Two Objective Pareto Set

Figure 6-14 shows the true two-objective Pareto set found via exhaustive search. There are a number of solutions on the plot that appear as though they are dominated, but in fact they differ in the n^{th} decimal place in at least one objective to their nearest neighbouring solutions.

The radiation pattern of some solutions had a mainlobe, but no sidelobes within the angular sampling constraints (Figure 6-13a). When such patterns were found, the maximum sidelobe level objective value was set to equal the lowest value of gain measured within the pattern sampling constraints. Following this methodology, thirty-one Pareto optimal solutions exist in the search space of 2^{21} solutions. A complete listing of the objective values associated with this chart is given in Appendix A.

The Pareto front is clearly non-linear, has regions of discontinuity, and becomes more densely populated with solutions as the mainlobe gain and maximum sidelobe level rises. There are also some concavities present that would cause problems for many

classical non-linear optimisation methods, and for aggregated or weighted-aggregated GAs. It should be noted that the Pareto optimal solutions reported here are not necessarily all usable. The lowest sidelobe solution (7.18dB gain, -101.7546dB) has a mainlobe so wide that no sidelobes are present – in this case the sidelobe level is the lowest reported gain across the sampling region.

One interesting property of the Pareto set which is often disguised when patterns are normalised is the significant reduction in mainlobe gain level as the sidelobe level falls. Quite often in the literature, patterns are normalised (particularly when optimising sidelobe performance) and do not show this property [56 , 59, 93].

Using the non-dominated sorting method first described by Deb et al [69], the first five non-dominated fronts were determined and are shown in Figure 6-15.

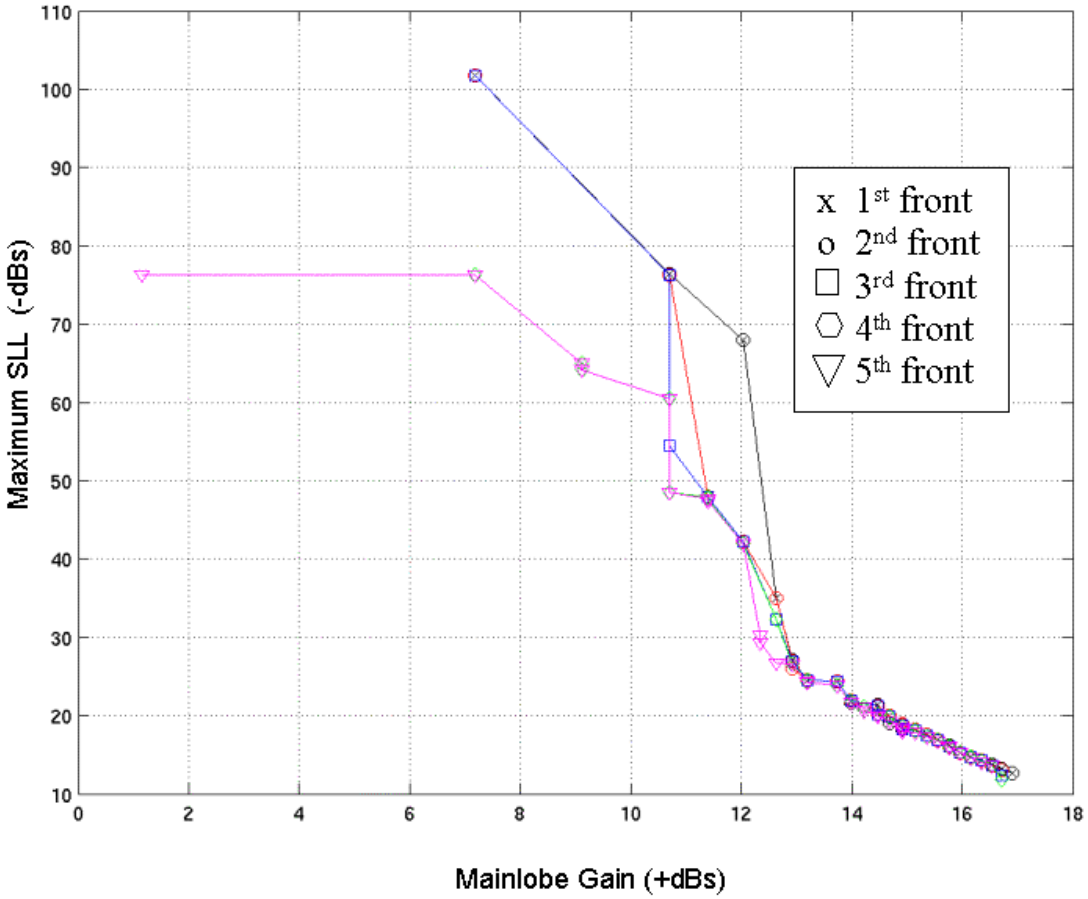


Figure 6-15 Sorted Non-Dominated Fronts for 2-Objective Problem

Figure 6-15 shows that the density of solutions with low sidelobe levels is much lower than for the higher gain patterns. Classical Monte-Carlo optimisers would have difficulty finding these solutions. There is also noticeable separation between the fronts in the lower sidelobe region. Only 146 out of 2097152 solutions exist with maximum sidelobe levels of -50dB or better.

Figure 6-16 shows the affects of quantising the excitation values on the solution density (i.e. all solutions lying in distinct columns). The Figure also gives indication of how densely populated the objective space is.

The difference in objective space between neighbouring low sidelobe solutions is very large compared with those solutions with higher gain. Due to their low frequency of occurrence, the solutions with the lowest sidelobe levels should prove difficult to find.

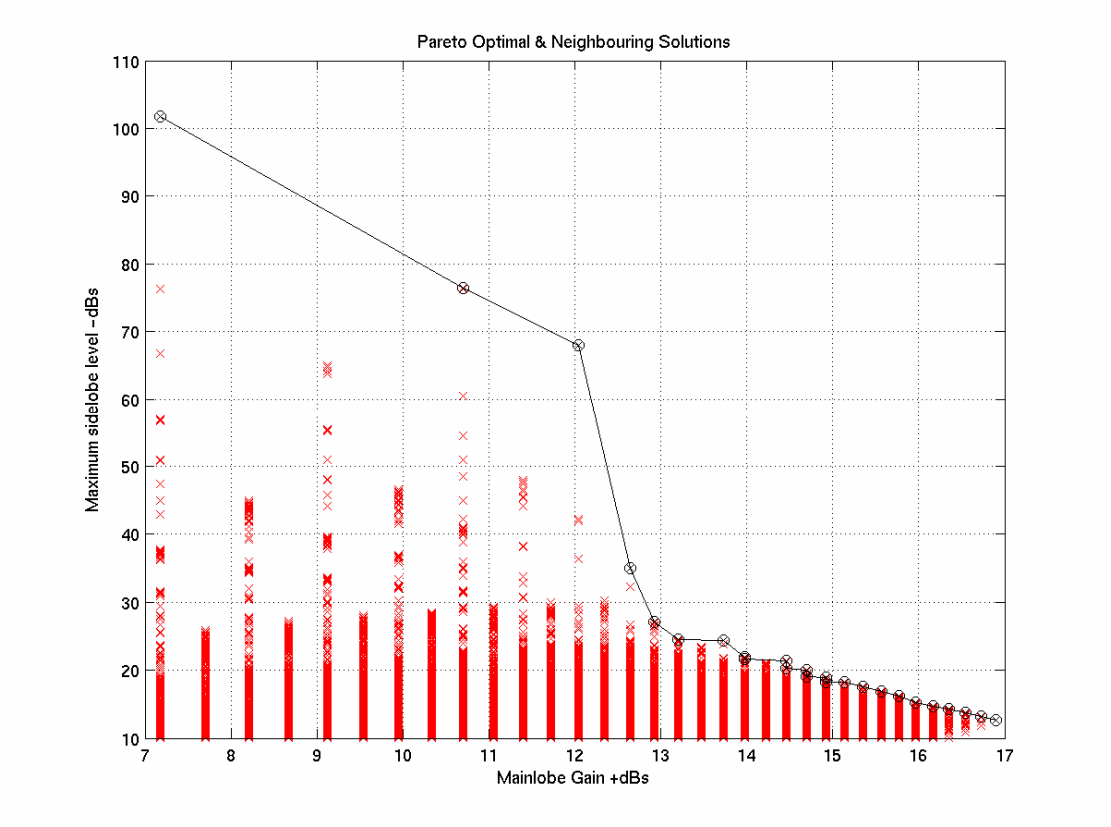


Figure 6-16 Solution Density

6.7.2. Three-Objective Pareto Set

The three-objective Pareto set shares many similarities with the two objective set. Again, the Pareto surface has regions of discontinuity, and becomes more densely populated as the mainlobe gain and maximum sidelobe level rises. There is a good spread of first sidelobe level values in the densely packed region of the surface.

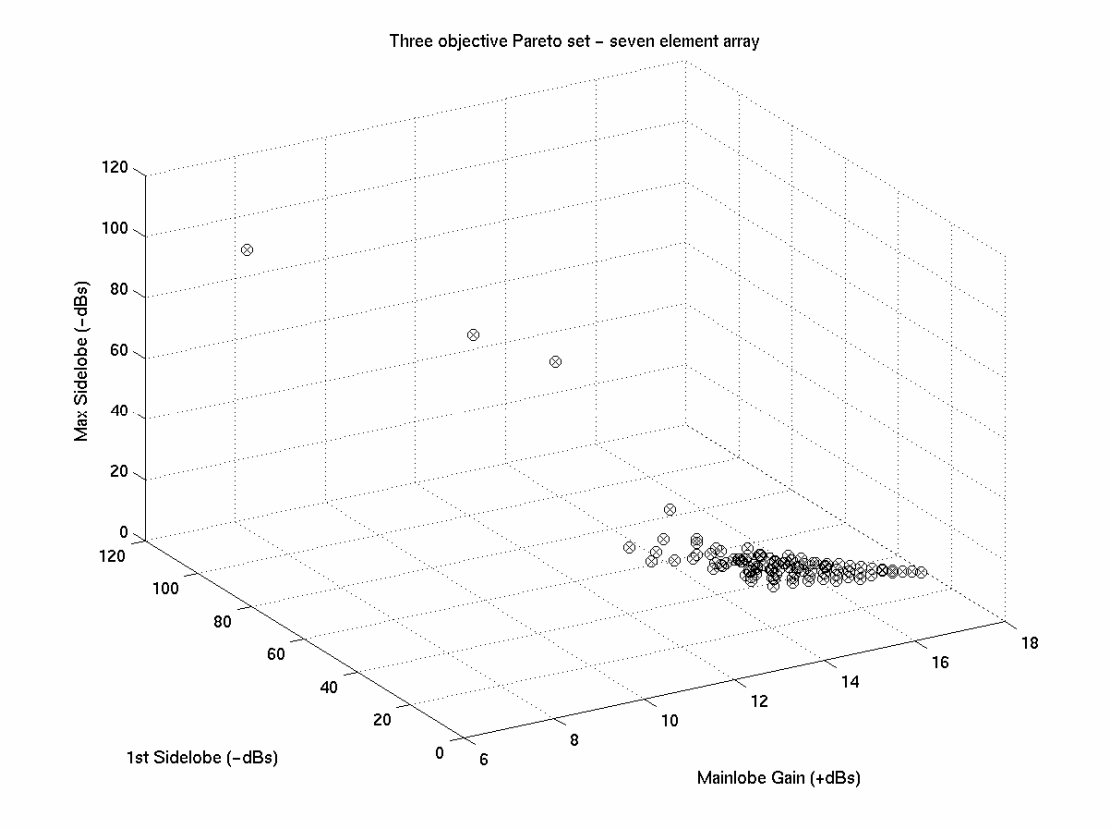


Figure 6-17 Three-Objective Pareto Set

The three-objective Pareto set contains 133 non-dominated solutions. As with the two objective results, the complete set is tabulated in Appendix A.

Both the two and three-objective Pareto solution sets have highlighted the essential need for the EA to offer good diversity performance in order to find the solitary low-sidelobe solutions.

6.8. EA Performance Evaluation

As a basis for further work, the performance of three recent EAs described earlier in this chapter have been evaluated against the new two and three-objective antenna test case problems.

The three algorithms evaluated (all described earlier) were the Strength Pareto Evolutionary Algorithm 2 (SPEA2), the controlled elitism version of the Non-dominated Sorting Genetic Algorithm 2 (c-NSGA2) and finally the Epsilon Multiobjective Evolutionary Algorithm (ε-MOEA).

The conditions for each algorithm were made consistent - each algorithm was set to evolve solutions for three hundred generations, with a population size of three hundred. Each algorithm evaluated 90,000 solutions, equivalent to searching 4.29% of the entire decision space. To account for the stochastic nature of EAs, each algorithm was run ten times. Probability of crossover was set at 90% and mutation at 0.5%. A uniform crossover operator and a single bit mutation operator were used in all cases. The

excitation was encoded in a single chromosome containing 21 bits (given that only seven elements are present, the SWS encoding scheme developed in Chapter 5 was not used here).

6.8.1. SPEA2 Two Objective Test Case

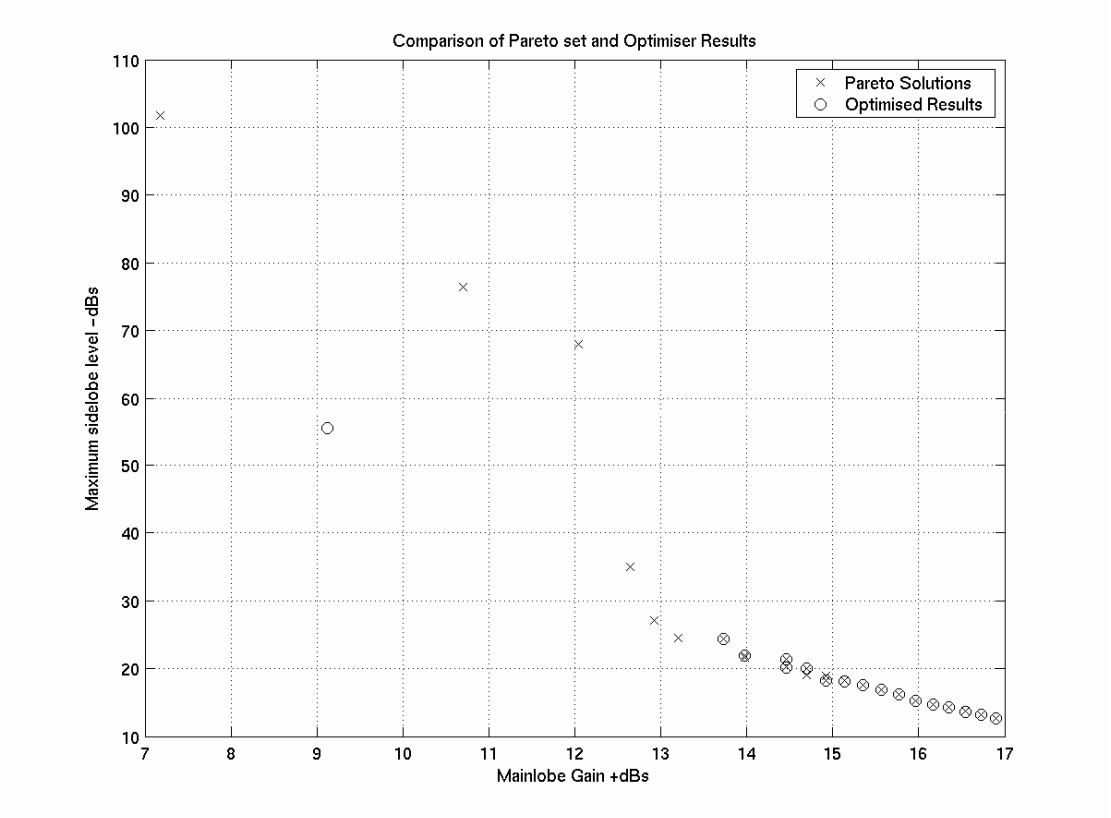


Figure 6-18 Result of a Single SPEA2 run.

The first run of SPEA2 ended with the solution archive containing eighteen non-dominated solutions. None of these solutions were exactly the true Pareto optimal solutions but as can be seen in Figure 6-18 the algorithm did especially well in finding solutions close to the higher gain values (differing in the second decimal place).

Ten runs in total were completed, and the best results from each run were used to form a combined set of solutions. This combined set was then sorted to remove any dominated solutions. Figure 6-19 shows where the combined results set lies in objective space and how close these solutions are to the true Pareto optimal set.

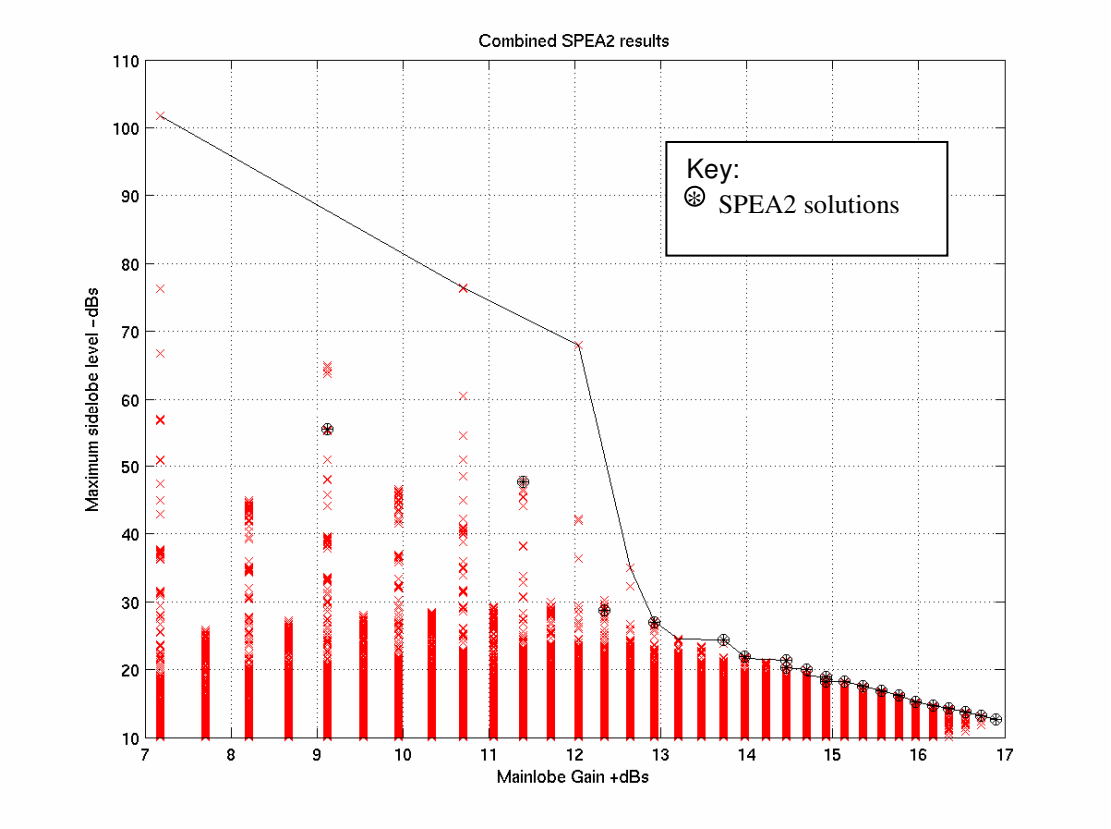


Figure 6-19 Combined SPEA2 Results Set

Against the two-objective problem, SPEA2 performed consistently and easily found solutions in the densely packed region of the Pareto front. While SPEA2 did not find any of the lowest sidelobe Pareto optimal solutions, it did find a usable solution with sidelobes better than -50dB.

6.8.2. SPEA2 Three-Objective Test Case

The performance of SPEA2 on the three-objective problem was very similar to that on the two-objective test case. SPEA2 successfully found high numbers of the higher gain solutions, but struggled to come close to the Pareto optimal sparsely populated low sidelobe solutions.

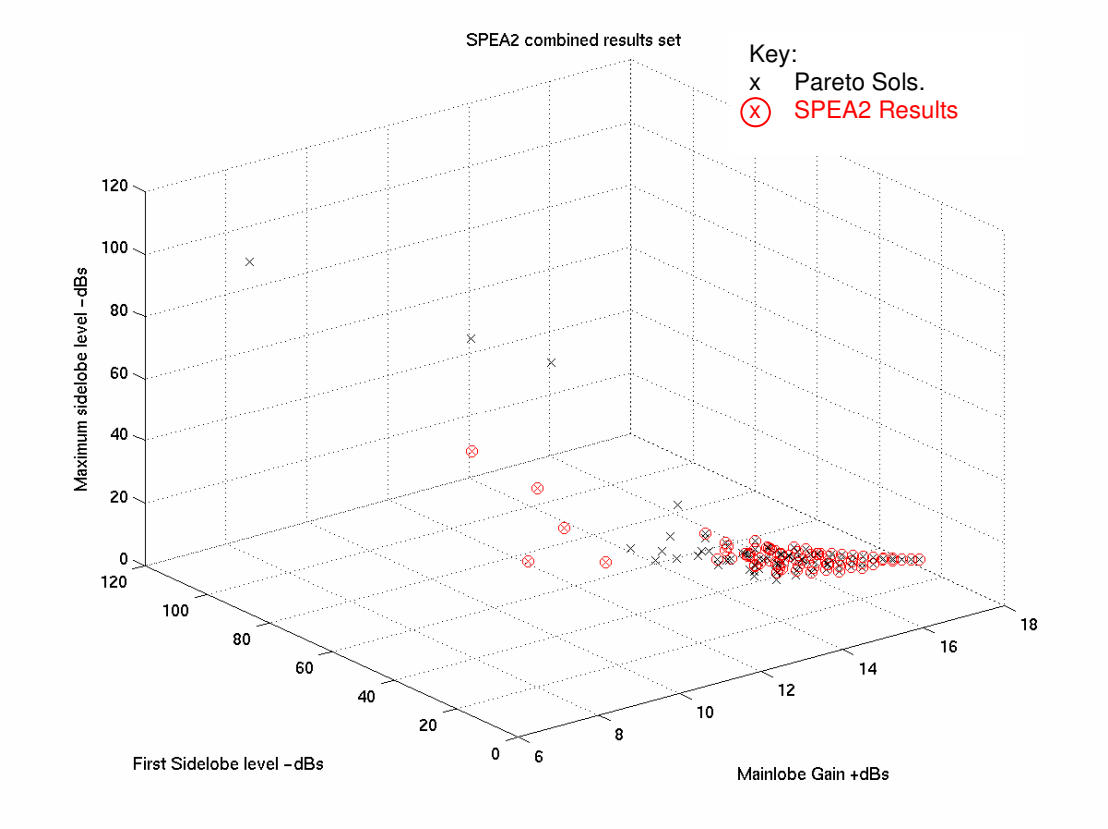


Figure 6-20 SPEA2 Results on 3-Objective Problem

The algorithm provided very usable solutions and provided a good approximation of the majority of the Pareto solution surface.

6.8.3. NSGA2 Two Objective Test Case

A number of problems were experienced when applying the controlled elitism NSGA2 algorithm to these test case problems.

The initial random population was successfully evaluated and a new population of child solutions formed. These too were successfully evaluated and the parent and child populations merged as specified in the algorithm. In the antenna test cases, the combined set of parent and child solutions contained many duplicate solutions.

When duplicate solutions existed in the combined parent and child population, the non-crowding distance calculation becomes invalid as crowding distances become equal to zero (i.e. the nearest neighbour is at the same point in objective space). The fact that probability of crossover was 90% ensured that at least a small number of solutions are retained from parent to child population forcing duplicate solutions to exist in a combined set. The calculation of rank was not affected.

Zero values of crowding distance had a secondary effect on the crowded tournament selection operator: the algorithm relies on a tournament selection operator that differentiates between solutions based on their rank and crowding distance (known as

crowded comparison criterion). The crowded comparison operator states specifically that when comparing two solutions *i* & *j*, solution *i* is better than solution *j* if:

$$(irank < jrank) \text{ or } ((irank = jrank) \text{ and } (idistance > jdistance))$$

In the event that two solutions exist in the population that belong to the same front and have the same fitness values (when duplicate solutions exist), the methodology fails.

Efforts were made to modify NSGA2, such as filtering solutions with duplicate fitness values, but it was found that there were not enough well performing new solutions at each generation to maintain the population size.

Comments contained in source code downloaded directly from the NSGA2's authors read "(sic)…maintaining a good distribution of solutions in problems having quite a different range of objective functions were difficult." [136].

Later work by the same author's makes further reference to the performance of the algorithm and suggests that the diversity in solutions achievable by NSGA2 is not expected to be as good as that achievable with SPEA2 [124] .

In order to make the algorithm work on the antenna test cases, a number of modifications to the basic algorithm were necessary:

- All duplicate pairs of solutions were identified, prior to the calculation of the crowding distances.
- The duplicate solutions were temporarily removed and the calculation of crowding distances made for all unique solutions.
- Each duplicate solution was then assigned the same crowding distance as its pair as identified earlier.

This method prevented zero-valued crowding distances from occurring, and presented the selection stages of the algorithm with more diverse information with which to guide convergence. The disadvantage was that duplicate solutions hold back the growth of diversity in the solution set.

Another potential problem with the methodology was with the controlled elitism operator. The algorithm description states that the controlled elitism operator ensures diversity is maintained in the population (across non-dominated fronts) by use of geometric function that allows a reducing number of solutions to exist in each front. For example, the geometric function may allow fifty solutions to exist in front one, thirty five in front two and so on so that only a small number of low ranking front members are maintained. The main problem is that the algorithm states that if there are not enough solutions within a population to fill the maximum permitted number of solutions in a front, the shortfall is added to the limit placed on the next front. Therefore if fifty solutions were allowed in front one, but only fifteen solutions (with rank equal to one) were present in the current population, a further thirty five solutions would be allowed to be maintained from the second front.

In the antenna test cases, there were rarely enough non-dominated front members present to fully occupy the permitted front and inevitably this fact lead to high numbers of solutions belonging to the first few fronts being maintained. Over many generations the convergence in the algorithm was limited due to a lack of diversity in the maintained solution set. This observation was confirmed by the two objective results (Figure 6-21).

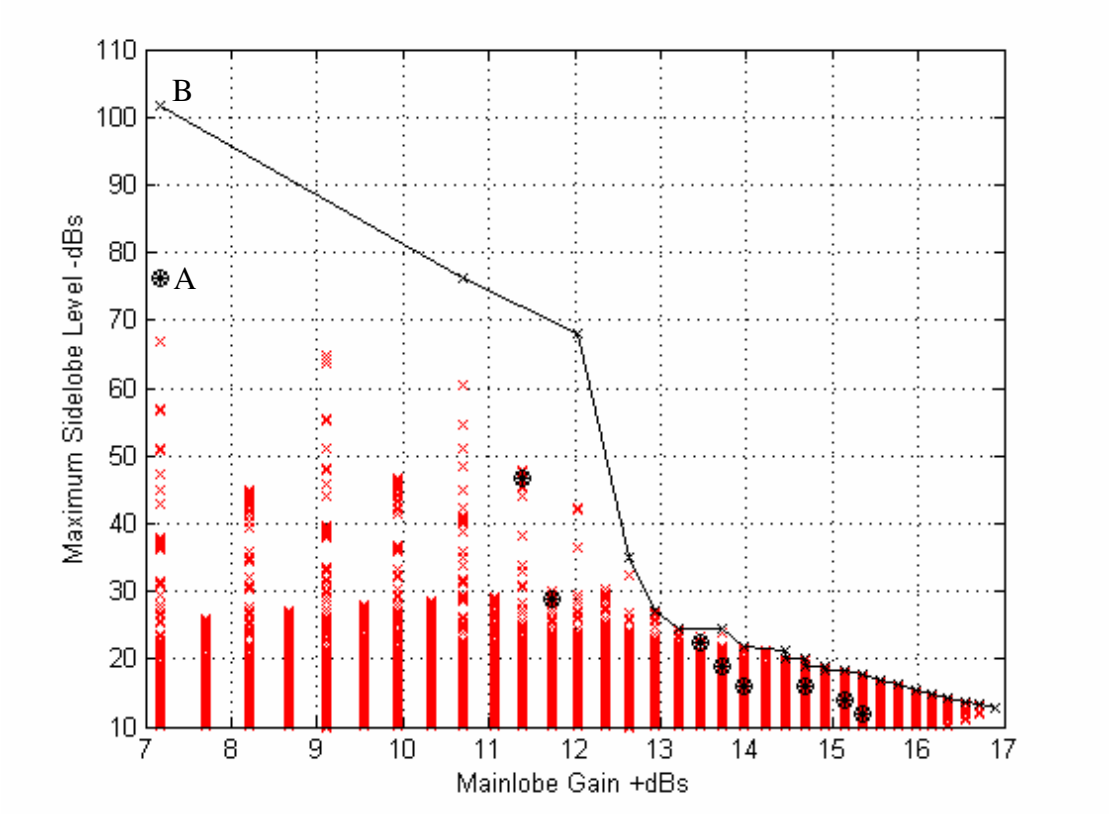


Figure 6-21 Controlled Elitism NSGA2 - Two Objective Results

Removal of the duplicate solutions prior to making the crowded distance calculations has helped to promote diversity; NSGA2 was the only algorithm to find the solution marked ‘A’ on the above chart, and this solution is the nearest neighbour to the lowest sidelobe solution ‘B’. The final combined set of non-dominated solutions contained only nine solutions, by far the lowest number of solutions found by any of the algorithms under test.

6.8.4. NSGA2 Three-Objective Test Case

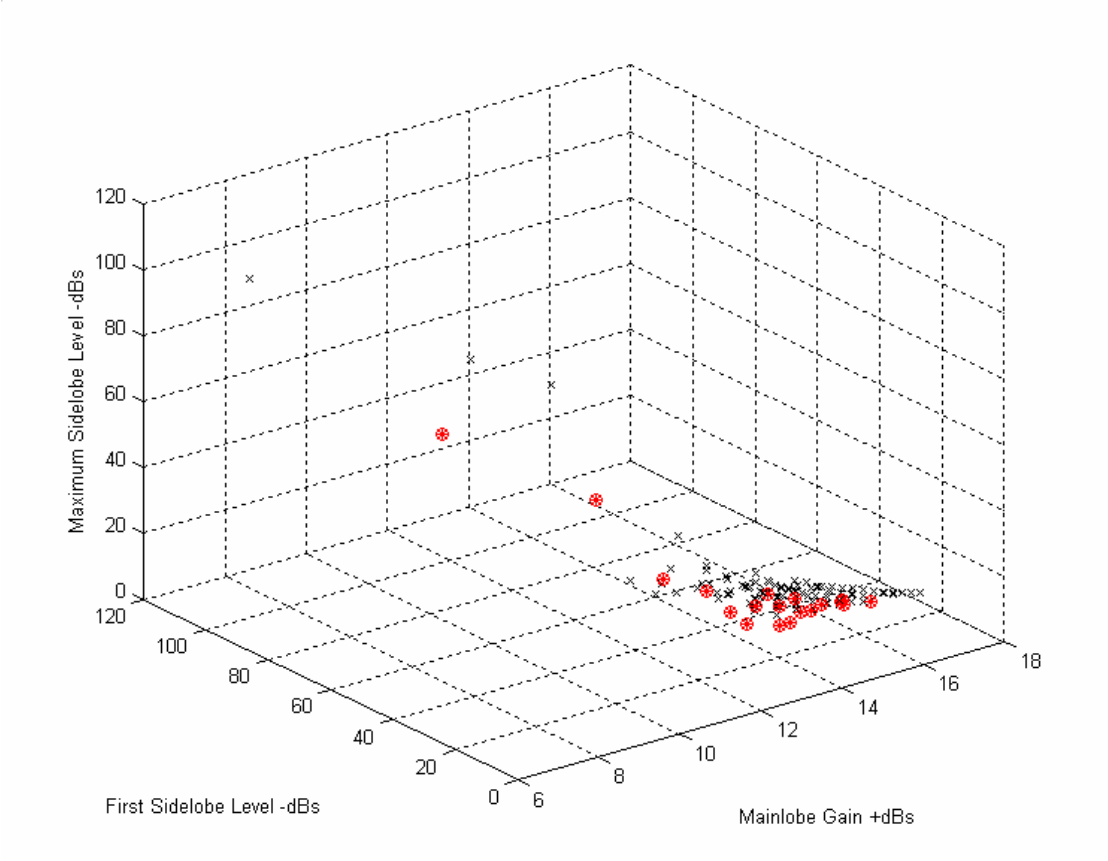


Figure 6-22 Controlled Elitism NSGA2 - Three-Objective Results

With the three-objective problem, NSGA2 failed to find any of the solutions in the densely packed regions of the Pareto set. A small number of lower sidelobe solutions were found that are perfectly usable, one of which belongs to the true second non-dominated front.

6.8.5. ε-MOEA Two Objective Test Case

The performance of ε-MOEA was outstanding on the two-objective test case. From the ten complete runs of the algorithm, Figure 6-23 shows the best set of results obtained.

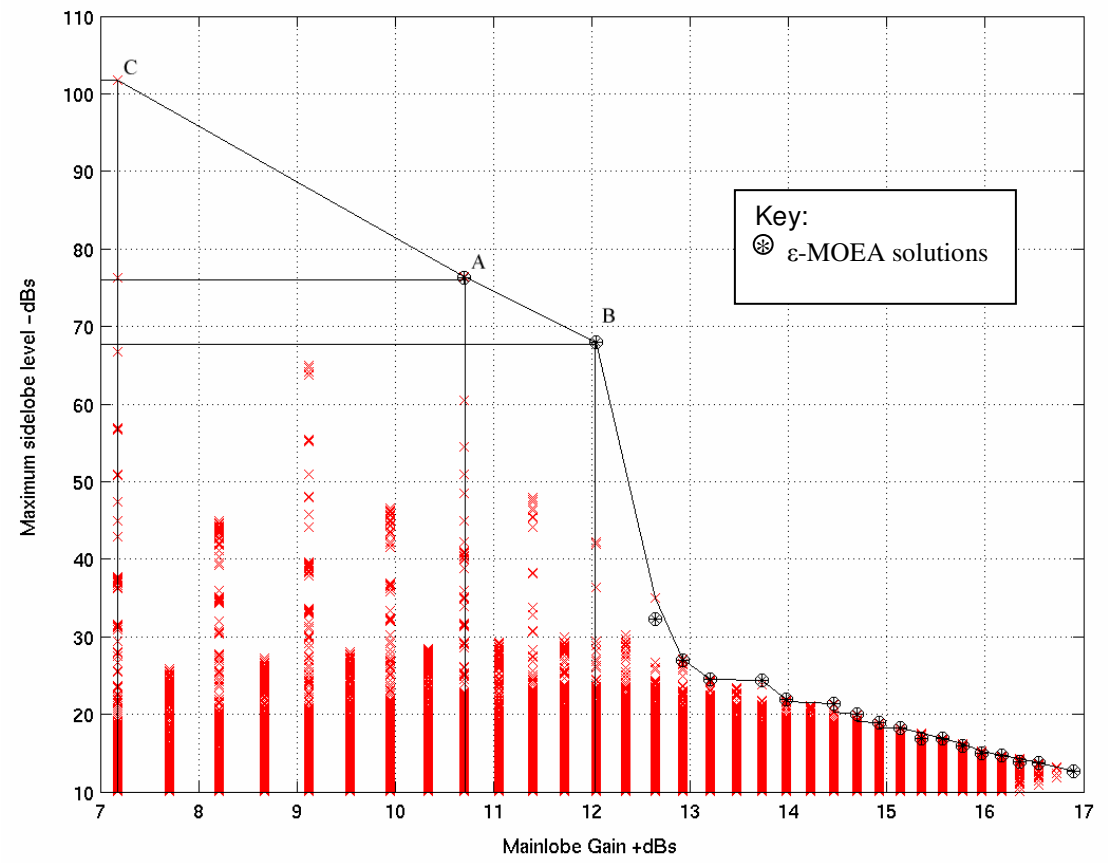


Figure 6-23 ε-MOEA Performance

On most of the ten runs of the algorithm, the final set of solutions was as good as the combined set from all runs.

This particular set of results highlighted a more general limitation of searching for non-dominated solutions. In Figure 6-23, the solutions marked (A) and (B) dominate all usable solutions as shown by the overlaid rectangular regions. While these solutions are dominated, they are still usable, and would give the decision-maker (DM) more 'good' solutions to choose from, particularly if gain is to be meticulously controlled.

The identification by ε-MOEA of solutions belonging to non-dominated fronts has advantages over SPEA2 as many of the solutions dominated by individuals A and B are usable (particularly when gain must be precisely controlled). Having knowledge of neighbouring non-dominated fronts is clearly of advantage in this problem.

To find the lowest sidelobe solution, the EA would need to be re-run and guided purely by the sidelobe level objective measure in order to find the corner of the Pareto set. This method is not particularly efficient but would increase the probability of finding 'needle in a haystack' solutions.

The solitary solution has an amplitude taper with two of the elements switched completely off. The reduced power level accounts for the low gain of the solution and possibly why the majority of other population members did not share this property (i.e.

exploration (mutation) may be required rather than exploitation of existing chromosomes).

Overall the performance of ε-MOEA was impressive and the computation time was an order of magnitude less than for SPEA2. The speed increase can probably be attributed to the fact that ε-MOEA is a steady state algorithm where all good solutions are maintained in the population and can influence selection contests immediately, speeding up convergence and reducing the amount of time spent evaluating poor performing solutions.

The division of objective space into cells (cubes or hyperboxes in the case of many objectives), does not appear to be a limiting factor in antenna optimisation, in fact E-MOEA forces the user to think about acceptable ranges for the solution set and uses this information to avoid maintaining high numbers of similar solutions. It also enforces diversity in the results set.

6.8.6. ε-MOEA Three-Objective Test Case

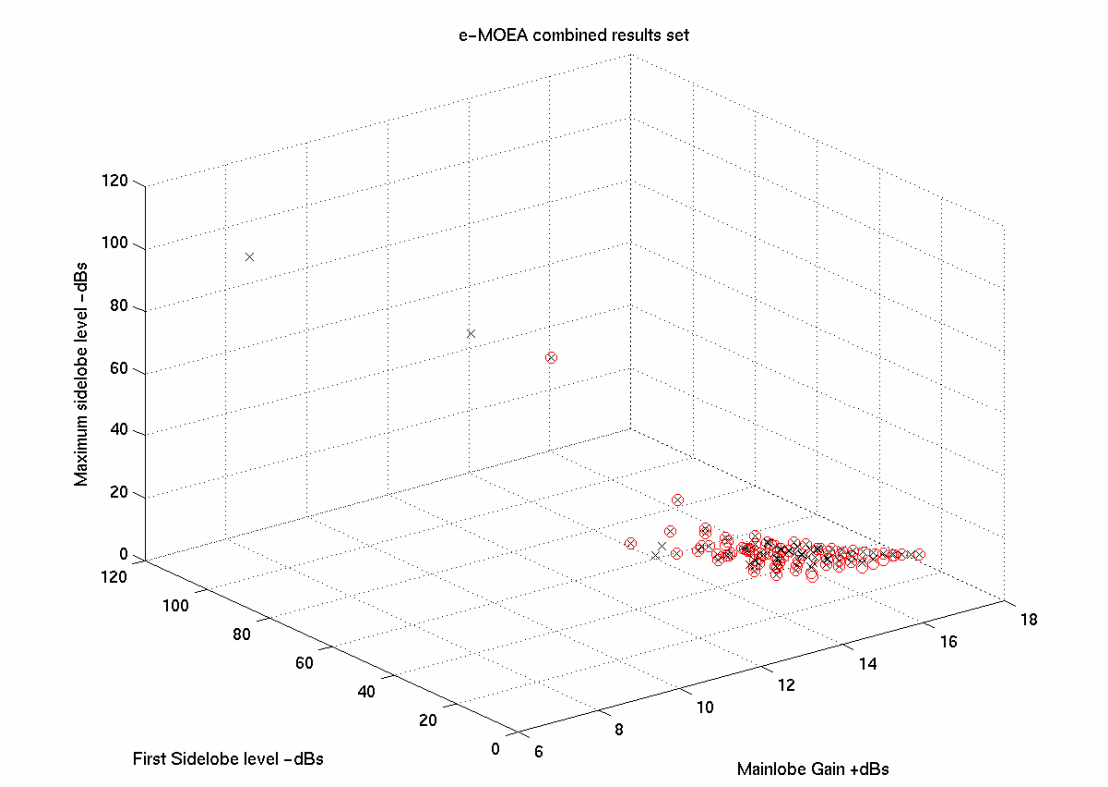


Figure 6-24 ε-MOEA Results on 3-Objective Problem

The affect of dividing objective space into hyperboxes can be seen in ε-MOEA's performance on the three-objective problem. The densely packed region of Pareto solutions are effectively thinned by the ε-MOEA results which is more practical from the decision maker's viewpoint and maintains diversity extremely well. The two lowest sidelobe solutions were not found, but it is possible to concentrate search in these

regions by reducing the number of hyperboxes that are used, e.g. removing all the hyperboxes that cover the densely populated regions of objective space.

6.8.7. Summary

The exhaustive search proved very informative and provided a good test case problem for evaluation of optimisation algorithm performance. A good multiple objective evolutionary algorithm must itself achieve a number of performance objectives. It should be computationally efficient, find solutions as close as possible to the true Pareto optimal solutions (high convergence) and provide a good spread of solutions to describe the Pareto optimal set (high diversity). Achieving these three-objectives means a compromise must be made.

SPEA2 had good convergence, but its diversity performance could be improved. It was also computationally intensive (particularly so when implementing its truncation operator).

NSGA2 presented good diversity, but poor convergence. Its computational time was low, so it scores well in that objective.

The recent ε-MOEA outperformed two other popular evolutionary algorithms and its performance excelled in computation time, solution convergence and diversity. The division of objective space into hyperboxes is a very practical and efficient means of improving algorithm performance and enforcing diversity in the solution set.

More generally, ε-MOEA has been demonstrated to be exceptionally good at the optimisation of multiple performance goals of linear array antennas. The algorithm found a very good approximation of the true Pareto front in a comparable computation time to that experienced during the application of simple GA to find a single solution by means of weighted sum cost functions.

One disadvantage of all these Pareto based methods is the computational time required to determine non-dominated solutions. The steady state ε-MOEA suffers in this respect, as it must compare each new child solution with both the archive solutions and the population members. Therefore for each and every cost function evaluation, a non-dominance check must be completed. By contrast, SPEA2 and NSGA2 are generation-based algorithms that only require a non-dominance calculation once per generation.

Also of note was the fact that as the number of objectives increased from two to three, the number of non-dominated solutions increases. If many more objectives were considered, the majority of the population is likely to become non-dominated. As many of the existing MOEAs rely on some kind of dominance measure to guide convergence, when the population is non-dominated, the only distinguishing parameters to guide the algorithms are the diversity measures. The end result would be a well spread population with poor convergence. It is essential that when using the evolutionary algorithms described above, that the number of objective measures is kept to the absolute minimum necessary to define the required radiation pattern performance.
When the number of objectives are less than say six, the problems with MOEAs can be mitigated somewhat by increasing population size, but new approaches are needed to

cope with many objective problems. Recent work by Purshouse and Fleming [137] studies MOEA performance on some generic many-objective problems, and reports similar findings.

CHAPTER 7
Array Partitioning

7. Array Partitioning

7.1. Subarrays

A common practice in the design and operation of antenna arrays is to partition their radiating elements into smaller groups known as subarrays. Partitioning makes possible the following RF functions:

- adaptive suppression of main beam or multiple jammers to reduce degradation in radar coverage,
- angular superresolution,
- antenna pattern shaping,
- forming clusters of beams for improved search times,
- forming sum and difference patterns,
- correction in the antenna pattern in case of failure of complete receiving channels.

Partitioning also helps reduce manufacturing costs by modularising array components.

Figure 7-1 (overleaf) shows an array architecture that allows full digital control at both the element and subarray level. Not all arrays have this full degree of control and many may be constrained by only having control at the subarray level. The reduction in control components reduces costs as a single phase shifter device can be used control an entire subarray receiving channel.

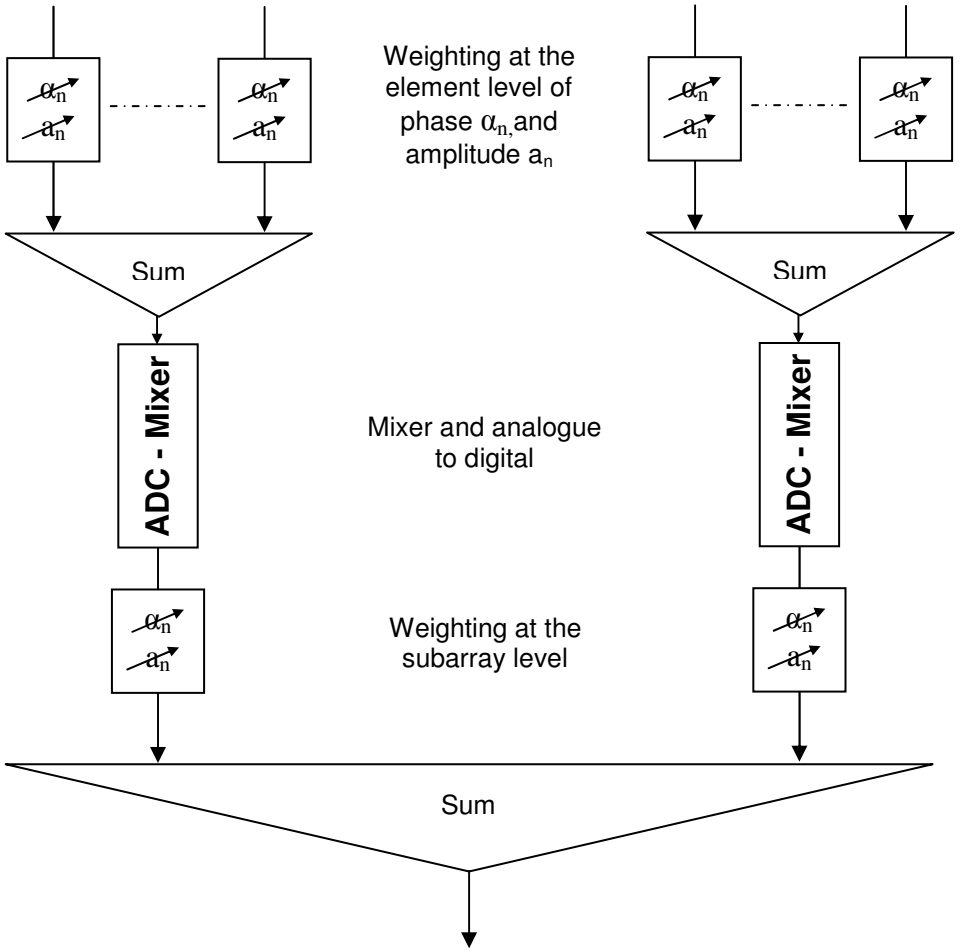


Figure 7-1 Architecture for Digital Beamforming in a Partitioned Array.

A current design trend in active array antenna technology is to move away from control at the subarray level and move to full control of the array at the element level. Control at the element level also means that 'virtual' subarrays can be formed during processing of the digital output from each TRM. These virtual subarrays could change according to RF function during the array operation. Achieving this control is expensive as all the RF signals must be digitised at the array face, but it does have the advantage of reducing signal loss. The architectural arrangement for an array of this type is shown in Figure 7-2.

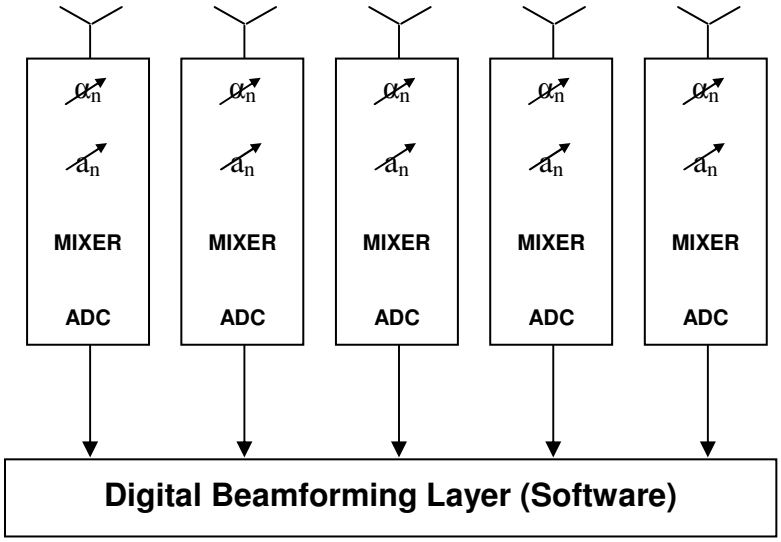


Figure 7-2 Fully Digitised Array

7.2. Approaches to Subarray Design

Linear array antennas can be partitioned quite simply by choosing a number of divisors within the array. The elements between each break point form the subarray grouping.

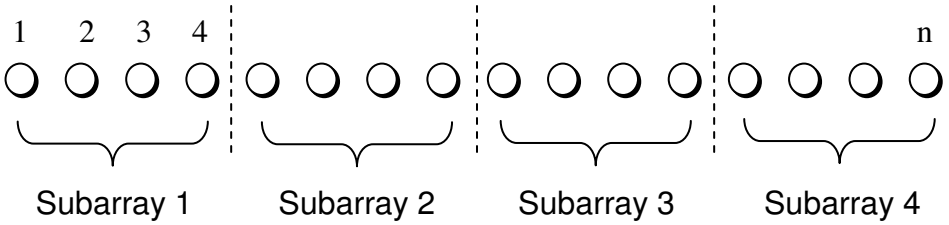


Figure 7-3 Linear Array Partitioning

It is reported in the literature that using regularly spaced subarray partitions can introduce undesirable grating lobes into the radiation pattern, and increase sidelobe levels [138,139]. It is also necessary to apply the correct excitations to the elements in each subarray. A common approach is to make random trial and error changes to the subarray partitions in order to search for a configuration that reduces these effects:

Nickel [140] partitioned a circular planar array using a trial and error method to find a configuration that produced a difference pattern with low sidelobes. The array of approximately 900 elements was partitioned into 32 subarrays and achieved a difference pattern with a maximum sidelobe level of -35dBs, but the same configuration did not result in a low sidelobe sum pattern. The amplitude weightings were based upon a Taylor series. The subarray partitioning has been reproduced in Figure 7-4.

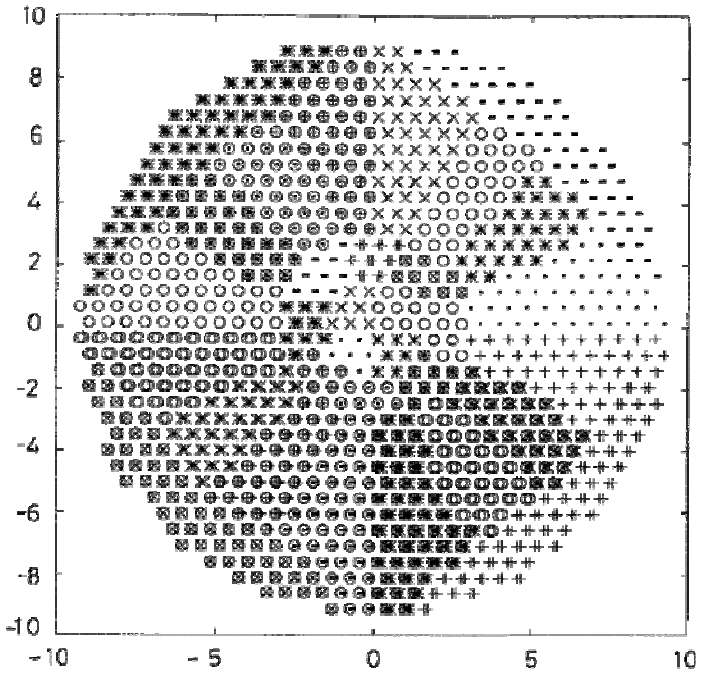


Figure 7-4 Subarray Configuration Derived by Nickel

Work by Tarran et el [141] investigated very large arrays (up to 10,000 elements) where the subarray partitions were completely randomised. They concluded that each subarray must be unique in shape in order to remove any periodicity from the design. Later work by the same authors [73] suggests use of regular shaped building blocks that are randomly combined to create an irregular set of subarray phase centres (see Figure 7-5).

Taylor weightings were applied to achieve low sidelobe sum patterns, and Bayliss weightings [142] used to give the difference patterns. The elements in the array were assumed to be isotropic. Each subarray in their suggested configurations contained roughly equal numbers of elements. No attempt was made to optimise the partitioning used; the configuration chosen was made at random.

In order to form the Bayliss difference patterns, a number of the subarrays were combined to reduce their numbers from 64 to 16. The Bayliss weighting was then approximated on the reduced set of subarrays. This weighting resulted in a raise in the maximum of sidelobe level of around 20dBs, compared with applying the weighting at the element level.

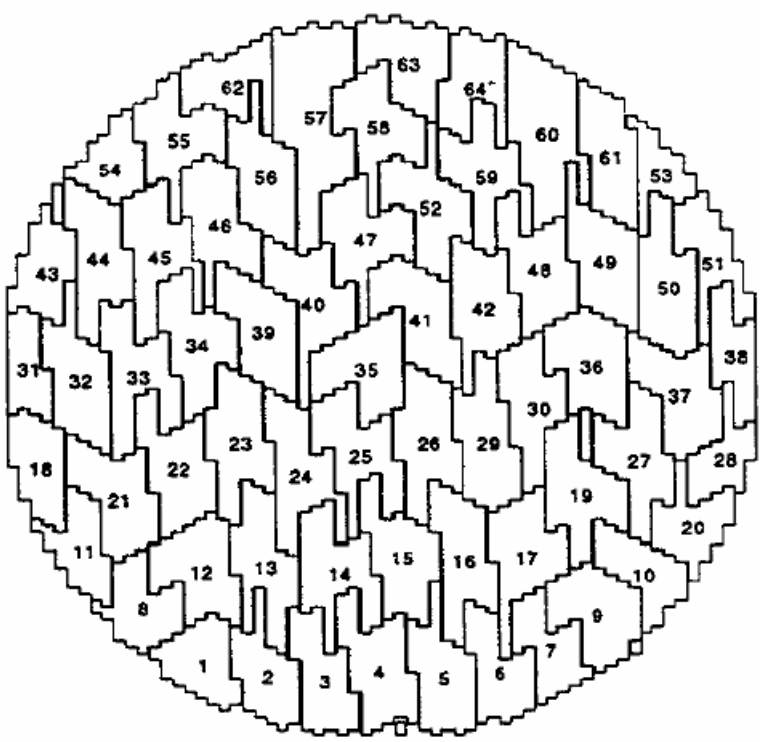


Figure 7-5 Subarray Configuration Derived by Tarran et al [76].

7.3. Subarray Optimisation

There are very few examples of specific attempts to optimise subarray geometry in the literature. It is possible to argue that the work cited above that uses trial and error is a form of Monte-Carlo optimisation, however due to the work involved in partitioning the array and evaluating the subsequent patterns, it is unlikely that many configurations were trailed. Much of the literature investigates optimisation of excitation sets of a fixed subarray configuration:

Alnajjar and Wilkes [143] minimised the sidelobe performance of a linear array through a trial and error procedure. Their procedure started with two small subarrays and varied the distance between them until the array performance improved. They then added a third subarray and continued. This resulted in adequate end performance, but the authors conclude that the final solution found was non-optimal. The authors suggest that planar subarray design could be performed by layering up the linear array solutions, although the method is not demonstrated.

Goffer et al [144] derived a closed form expression for the radiation pattern of a randomly partitioned linear array, although no optimisation is performed or suggested. Similarly Zhiyong et al [145] partitioned a linear array using a trial and error method to give good sidelobe performance and demonstrated how weighting at the subarray level can improve Space Time Adaptive Processing (STAP) performance.

Smolko optimised a linear array antenna that had already been divided into a number of regular subarrays [146]. Excitation was chosen at the subarray level using Woodward synthesis [8]. The regular arrangement of subarrays was shown to bring grating lobes into the radiation pattern, and it was shown that these lobes could be reduced by optimisation of the subarray weights. Weighting the subarrays appropriately could also produce flat-topped beams.

7.4. Subarray Optimisation using Modern Techniques

Early work by Haupt [61] in 1995 optimised subarray amplitude tapers, but the subarrays were fixed. Haupt used a genetic algorithm to search for uniform excitation values to apply at each subarray (i.e. all elements within the array were uniformly weighted). Both planar and linear arrays were optimised, but the subarray partitioning was not subject to optimisation. The subarray shapes in both planar and linear arrays were symmetrical.

Genetic Algorithms were used by Wang, Israelssson and North [147] to optimise subarray configurations in a linear seismic array. The array operated at low frequencies and the elements in the array were spaced several kilometres apart. The GA optimised a binary string with one bit for each element. The bit state represented whether the element was switched on or off. When used in the manner, the GA was actually optimising array thinning rather than subarray partitioning as the paper title suggests. The authors suggest that the chromosome could be lengthened to include amplitude weights.

Ares et al [148] used simulated annealing to optimise a number of parameters in a combined fitness function on subarrayed linear and planar antennas. Again the subarrays were fixed, but objectives such as minimisation of the difference between maximum and minimum weights were included in the fitness function. The work produced weights that provided good sum and difference patterns from the same array configuration.

López et al [74] used a simple genetic algorithm to optimise both the partitions and the weights applied to a linear array. The number of subarrays was fixed and not subject to optimisation. The cost function aimed to minimise a single objective that relates to maximum sidelobe level. Mutation was applied to every chromosome in the population and may have affected convergence of the algorithm (earlier work in Chapter 5 showed little convergence improvement or improvement in the final solution when the mutation rate was raised beyond 5%). The type of mutation operator used was not specified.

Golino [149] used a simple genetic algorithm to optimise the performance of a partitioned array being subjected to electronic countermeasures (ECM). The array was small (a 64 element planar array) and the GA was set so that a triangular region of the array could share its elements with four other subarrays. While some resultant partitioning arrangements are presented, no results are given on how the array configuration performed.

7.5. Multiple Beams

A partitioned array can be used to form multiple beam patterns. Multiple beams are required by many in modern multifunction radar and communication systems as they offer numerous benefits, examples include:

- angular resolution can be improved by using a cluster of narrow pencil beams to cover the same search area as a broader beam,
- energy management (closer targets can be illuminated with a lower gain beam),
- frame-time during search functions can be reduced.

The use of multiple beam antenna patterns is of great benefit in active array communication satellite systems. This beam-shape flexibility greatly reduces the business-plan risk associated with the satellite, because it is possible to reconfigure the coverage area to respond to changing markets. This flexibility is particularly true for satellites with a large number of independent beams because each beam can be tailored to the specific coverage area required by its customer [150]. There are problems with some existing methods of forming multiple beams, in terms of controlling grating lobes in partitioned antennas, particularly when the array is steered.

Multiple beam formation in a partitioned array is of great use and it is thought that modern optimisation techniques can provide a means of finding the partitioning configuration and weightings to form such patterns efficiently.

7.6. Sum and Difference Patterns

When a single antenna is to be used for search and track functions as in many radar applications, the antenna must be capable of producing a sum pattern (where a single mainlobe is produced with low sidelobes elsewhere) and a difference pattern (where two mainlobes are produced, close together with a deep null separating them). The multiple lobes in the difference pattern allow detected targets to be accurately located in azimuth and elevation (two difference patterns are required for both azimuth and elevation location). Again with the difference pattern, low sidelobes are required.

The excitation sets required for sum and difference patterns are dissimilar. Once an array has been partitioned, a compromise between good sum and difference pattern performance is necessary. McNamara [151] chose to find an optimal sum pattern performance for a linear array, and then compute subarray weights to give a best compromise difference pattern.

Mohamed and Holden [152] minimised the grating lobes that appeared in a difference patterns of an array with fixed subarray partitions. Their method produced good patterns with both sum and difference patterns having sidelobes -36dB relative to the main beam. The authors state that the difference pattern performance was achieved at the expense of a higher first sidelobe level in the sum pattern. These findings are commensurate with the optimisation method used - a simple hill climbing optimiser.

7.7. Summary

On the whole, the subarray partitioning literature largely uses a randomisation technique to determine the subarray geometry, and a small number of configurations are trailed.

The excitation sets used are applied at the subarray level, and it is shown that some sidelobe level reduction and grating lobe control can be found by varying the subarray partitions.

The application of GAs in array partitioning has been limited but shows promise but the joint optimisation of sum and difference patterns from an array of fixed partitioning is not a mature technique and all prior work considers weighted sum type fitness functions. The algorithms provided a single solution to the problem and did not take into account the effect of the element radiation patterns.

The work by López et al was the only work uncovered that used GAs to actively search for a partitioning configuration that worked to improve radiation pattern performance. Their method was limited to linear array antennas and would not work on planar (2D) arrays.

There is clearly a need for a fast partitioning method suitable for use on planar arrays that can be subject to iterative optimisation algorithms such as EAs. Applying a uniform excitation level to each subarray will also reduce the size of the search space and should improve the optimiser's performance.

It is important that the partitioning and optimisation method should strive to be independent of element count, and ideally be applicable to arrays of all geometric shapes.

The optimisation of multiple-beam patterns, particularly where steered beams are required is worthy of further attention and should help an end user exploit the capabilities of expensive array antennas.

It is thought that optimal compromise between sum and difference patterns formed using a partitioned array could be better tackled with a multiple objective optimiser. Given the number of objectives that must be monitored, prior work that has used weighted sum cost functions and/or classical optimisation techniques has reported undesirable characteristics in the final radiation patterns.

The next chapter addresses these requirements and develops methods for partitioning arrays that could be used for current generation designs that fix subarrays during manufacture, or next generation digital arrays that can vary their partitioning during operation and hence offer maximum operational versatility.

CHAPTER 8
Partitioning Methods

8. Partitioning Methods

8.1. Introduction

Chapter seven discusses a number of different approaches for partitioning linear and planar array antennas. In order to apply iterative optimisation algorithms to determine a well performing configuration, the method for performing the partitioning needs to be formulated. The partition method must be able to create varied partitioning configurations at random in order for the optimisation algorithm to perform an efficient search of the design space.

It is important for the partitioning method to be orthogonal (i.e. the same input always gives the same output), fast and efficient. The number of variables required to encode the array partitioning must be minimised or it will add complexity to any optimisation process. A good partitioning method offers a good sampling of the design space with the least amount of variables required to encode the partitioning information.

Finally it is essential that the encoding schemes remain valid once subjected to genetic operators such as crossover and mutation.

8.2. One-Dimensional Methods

8.2.1. Simple Dividers

The partitioning of linear arrays is relatively simple to achieve. Methods exist in the literature such as the work by Lopez et al [74], although they do assume a fixed number of subarrays are required. Lopez and colleagues used a genotype that permitted non-neighbouring elements to belong to a subarray. This method would make the feed network especially complicated, particularly on larger arrays. In addition, a variable (needing optimisation) is required for each element in the array in order to assign it to a particular subarray. The authors suggest that use of a simpler method that optimises subarray boundaries would be better, but they do not suggest how to implement the idea.

Consider a forty-four element linear array that is to be divided into ten subarrays. Typically, when simple dividers determine subarray boundaries, the partitions can be

encoded in the genotype in a number of ways. One method uses two chromosomes to encode the partitioning and the weights as follows:

1. A single chromosome ch_1 is generated that contains Q values between zero and 0.3 (where Q is the number of required partitions, equal to ten in this case).
2. The variables within ch_1 are decoded such that each successive value represents a percentage of the remaining number of available elements in the array. The upper limit of 0.3 prevents more than 30% the elements belonging to the first subarray.
3. These percentages are rounded and the pro-rata numbers of elements are assigned to each subarray.

For example if ch_1 contained the variables:

$$ch_1 = [\ 0.05 \quad 0.15 \quad 0.19 \quad 0.15 \quad 0.18 \quad 0.10 \quad 0.22 \quad 0.04 \quad 0.16 \quad 0.30\ ]$$

the first variable (0.05) would imply that subarray one contains three elements (the first 5% of the forty-four elements within the array). The second variable (0.15) implies that subarray two contains seven elements (15% of the remaining forty-one elements) and so on. Once fully decoded the subarray partitions become of size:

$$S = [\ 3 \quad 7 \quad 7 \quad 5 \quad 4 \quad 2 \quad 4 \quad 1 \quad 2 \quad 9\ ]$$

This encoding method ensures that ch_1 is robust to crossover and mutation operators when used in a genetic algorithm.

A second chromosome ch_2 is used to encode ten amplitude weights, one for each of the subarrays.

8.2.2. Modified SWS – Random Signal Method

A second, alternative method is based on the SWS encoding scheme introduced in Chapter five.

If a random signal is created with a number of frequency components, properties of the signal can be used to not only partition the array, but also to assign the amplitude weighting to each partition. This new procedure is efficient, and is independent of the number of elements in the array. The method does not require an a priori value for the number of subarrays. Consider Figure 8-1 which shows a number of sine waves of different frequencies, phase shifts and amplitudes.

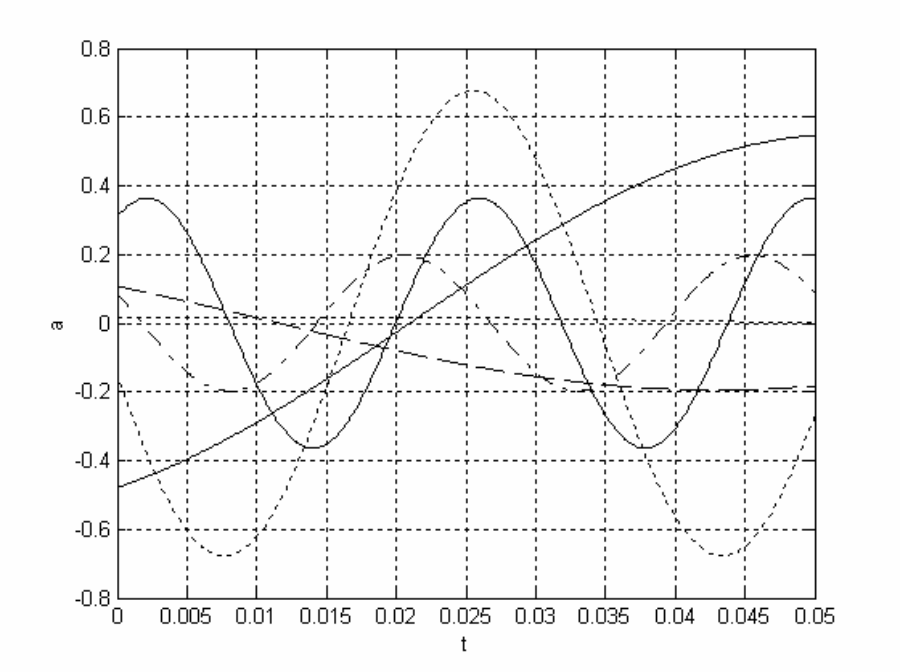


Figure 8-1 Sine Wave Components

If these sine waves are summed, a random signal is created as in Figure 8-2. This random signal can be analysed in much the same way as a radiation pattern, to determine the location of its peaks and nulls.

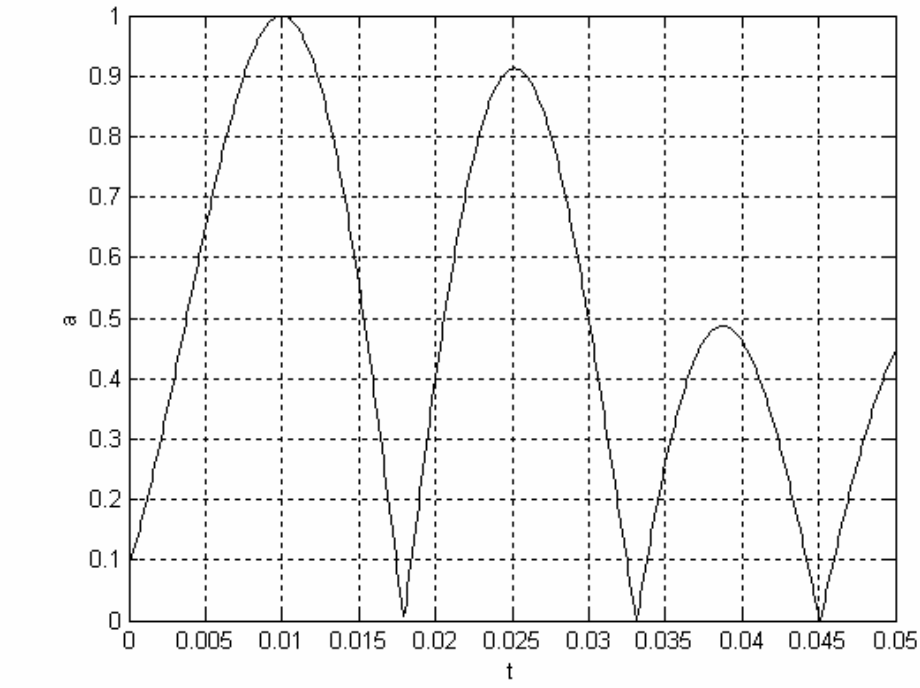


Figure 8-2 Sine Wave Sum

If the array geometry is overlaid proportionally along the *t* axis, the nulls can be used to represent the partition points and the magnitude of each lobe, the weighting to apply to the neighbouring subarray.

In terms of optimisation variables, at least six sine waves were found necessary to give a good variation in the number of partitions generated. Each sine wave requires an amplitude, a frequency and a phase shift variable to be set. In total, eighteen variables require optimisation, two less than for the simple division method, and this value does not increase with array size.

In Figure 8-2 each component sine wave was summed, squared and square rooted to ensure all the amplitude weightings were positive. By omitting this step, negative weights are introduced that can be used to represent phase value inversions for the generation of difference patterns.

8.3. Two-Dimensional Methods

Partitioning in two dimensions is more complex and neither of the methods above lend themselves particularly well to extension to a second or third dimension.

Earlier work such as Nickel in [129] based the partitioning shapes on weighting levels such as the Taylor. Given that the Taylor weighting is 'bell shaped' the subarray partitions must be formed into concentric rings around a centre element to approximate the taper. The corresponding amplitude weighting reduces in magnitude on each ring as they approach the edges of the array. Each ring was divided into a number of smaller partitions that could be weighted separately to form the difference pattern. While this method results in a compromise between sum and difference patterns, the partitioning was done 'by hand' - too time consuming for use in an iterative design process.

8.3.1. Random Signal Method

The random signal method can be used for planar arrays if concentric ring-shaped subarrays are desirable. In this case, the 'random signal' would be overlaid from the centre of the array to its edge to describe the radius of each ring boundaries (Figure 8-3).

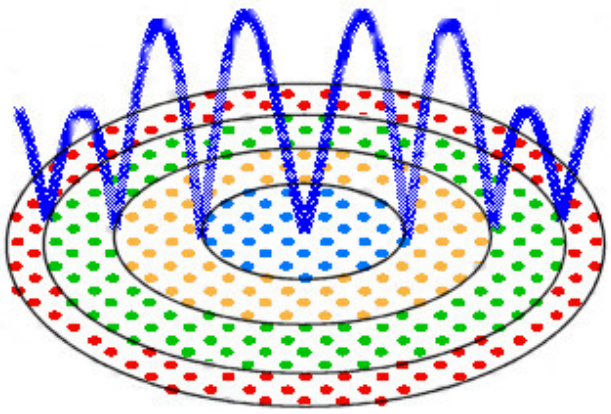


Figure 8-3 Random Signal Used for 2-D Partitioning

8.3.2. Binary Fission Algorithm

It is known from prior work, that introducing irregularity into the partitions reduces the magnitude of grating lobes in the radiation pattern. In a digitised radar, where all beamforming is performed in software, the subarray shapes can be made quite diverse.

Encoding the partitions efficiently is challenging and a 'random' subarray generator that can produce a diverse range of two-dimensional subarray shapes is very desirable.

The partitioning method needs to assign neighbouring elements to subarrays in order to create definite boundaries between subarrays.

After some experimentation, a new process was formulated that fulfilled all these requirements and encoded the variables associated with the process in a manner suitable for genetic optimisation.

In much the same way that the model of natural selection and survival of the fittest inspired the original creators of the genetic algorithm, the new partitioning method was inspired by the natural process of binary fission in cell division. Binary fission is described in many text books, for further reading refer to [153]. Put simply, binary fission is the process of a biological cell (called a mother cell) dividing into two daughter cells.

Bacteria spread by the process of binary fission and if one watches their growth under a microscope, they will appear to quickly cover a two-dimensional area. If a number of different types of cells, are placed close to each other in a Petri dish, and each cell type spreads at a different rate, at a time t later each cell type will occupy a unique two-dimensional region.

The process of binary fission as described can be modelled in an algorithm and it has many potential uses. A few simple rules are required such as constraining the spread of each cell type to a regularly spaced grid and preventing different cell types overlapping.

By making the analogy of the Petri dish being an antenna array face, then the array elements become possible locations for cells as they spread. Each different cell type represents a unique subarray. The properties of the cells as they spread meets the desirable characteristics for subarray shapes i.e. irregular, and that all subarray elements should be clustered together (not distributed across the array face). The desirable properties associated with this process are:

1. All the elements within a subarray are contiguous, i.e. physically located next to one another, which allows implementation in antenna arrays of all types of architecture.
2. The subarrays generated by the method are of reasonably uniform size and are thus readily possible to manufacture.
3. A relatively low number of variables are associated with the partitioning process, which permits an efficient binary or real valued string representation. This efficiency is advantageous for optimisation using genetic algorithms or other techniques.

4. The process can be applied easily both to planar array geometries and to conformal array geometries.
5. The process is suitable both for arrays, which are to have their subarray partitions fixed at the manufacturing stage, and for fully adaptive arrays, in which the partitions may be configured and modified in use according to function.

The biological process can occur at high speed. This process is an ideal candidate for software modelling and subsequent use in an iterative optimisation algorithm.

Two versions of a 'binary fission segmentation algorithm' were developed (subject to Patent applications as referenced in Chapter 1); the later version required fewer variables to describe the process. A summary of the general technique is first presented:

1. Form a grid representing the location of array elements;
2. For each subarray select an initial grid location for occupation by a theoretical biological seed cell of a given cell type representing that subarray;
3. Simulating a process of cell replication in which cells of a given cell type are associated with at least one corresponding growth rate variable and in which each existing cell is arranged on reaching a trigger size to generate at least one new cell of the same cell type. The new cell(s) are allocated to at least one pre-determined available new grid location adjacent to the respective grid location occupied by the parent seed cell;
4. When all the grid locations are occupied, identify the resulting boundaries between different cell types. These boundaries represent partitions corresponding to the said boundaries to the array to partition the elements into subarrays.

The steps in the process are shown in Figure 8-4.

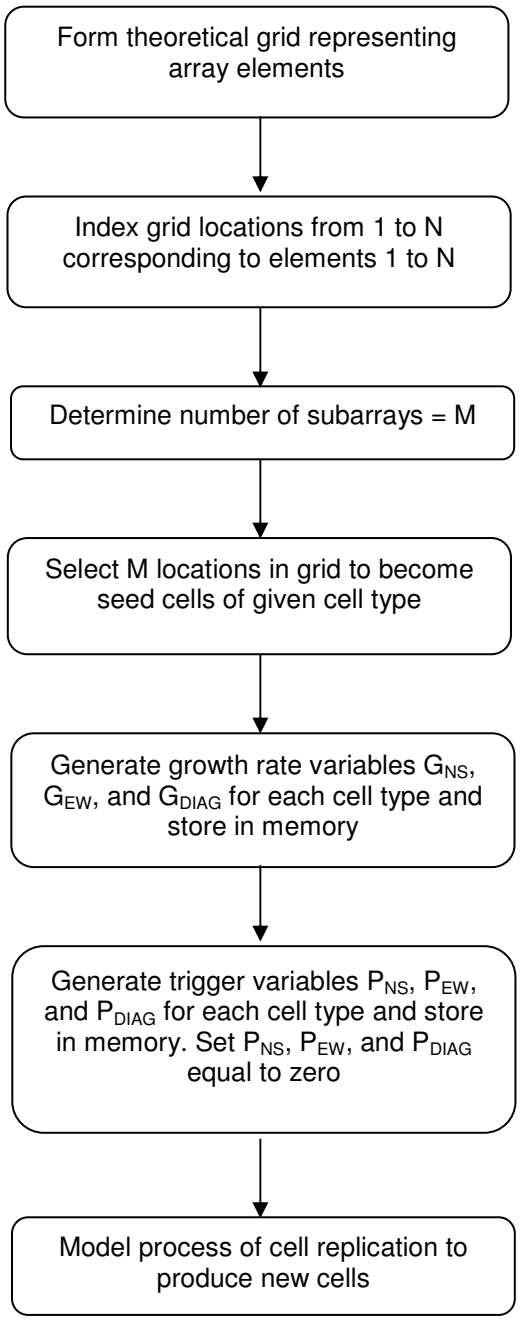


Figure 8-4 Binary Fission Algorithm

The pseudo code for algorithm implementation is given below. The algorithm is described with reference to a 15 x 15 element planar array antenna.

8.3.2.1.Initial Cell Placement

Described first is the procedure for the random placement of the various seed cells:

1. The first step is to create an index *I* of initial length N, which represents the grid locations indexed from 1 to N using whole integer numbers.
2. Next, select M random numbers between the values of zero and one (where M represents the number of subarrays required).
3. A string *ch1* is created containing these M numbers expressed as decimal values from 0.00 to 1.00. The values in the string *ch1*, when decoded, will act as pointers to positions within the index *I* and hence to locations within the grid.
4. I_{size} is set equal to the initial size of *I* (ie to the value N) and the first of the $1 <= i <= M$ values within the string *ch1* is selected.
5. Then, for this first and each of the subsequent value of the $1 <= i <= M$ values within the string *ch1*:

   Set seed location $i = (ch1_i) * I_{size}$.

   Set seed location i = round (i)

   If seed location i becomes equal to zero, set seed location $i = 1$

   The seed location will therefore be stored in position i of string *I*. The next step is to remove the value stored in position i of string *I,* hence reducing its length.

As successive seeds are chosen, therefore, the length of *I* reduces until it is of size N-M. The next steps involve updating I_{size} to be equal to the current size of *I* , and enquiring whether this size is equal to N-M. If yes, the procedure is terminated because M seed locations have now been chosen, corresponding to M subarrays. If no, the process proceeds to set $i = i + 1$ before returning to the start of step '5.' to repeat the process of determining the next seed location in the grid.

Example

First seed cell location:

I = [1,2,……,117,……225]

$ch1$ = [**0.52**, 0.43,…….0.62]

I_{size} = 225

Seed location i = round (0.52 * 225) = 117

$seed\_location_1 = I_{117} = 117$

I_{117} = [ ]; (i.e. is removed)

Second seed cell location:

I = [1,2,……225]

ch1 = [0.52, **0.43**,…….0.62]

I_{size} = 224

i = round (0.43 * 224) = 96

$seed\_location_2 = I_{96} = 96$

I_{96} = [ ];

8.3.2.2.Cell Spread

At this stage, all seed cells have been assigned positions on the array face. The second part of the description details how the seed cells spread to describe the subarray partitions:

With reference to the flow chart (Figure 8-4), a set of three growth rate variables G_{NS}, G_{EW} and G_{DIAG} is next randomly created for each one of the grid locations i_1 to i_M, ie for each cell type. The set of growth rate variables thus generated for each of the grid locations is stored in memory.

More particularly, now that the locations of the seed cells on the grid are known, three more strings *ch2, ch3* and *ch4* (each of size 1 x M) are created to contain the growth rate variables for each cell type. Each cell originally placed in the grid or subsequently formed, has three growth rate variables, determined by the cell type.

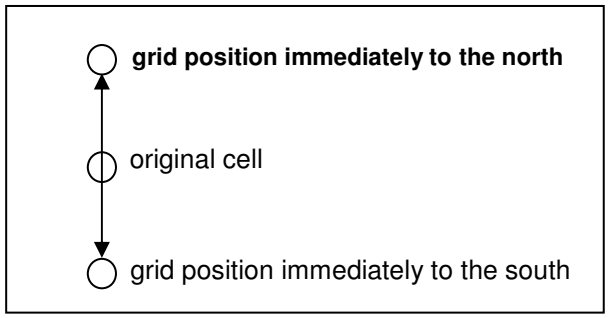


Figure 8-5 North and south growth rate, G_{NS}

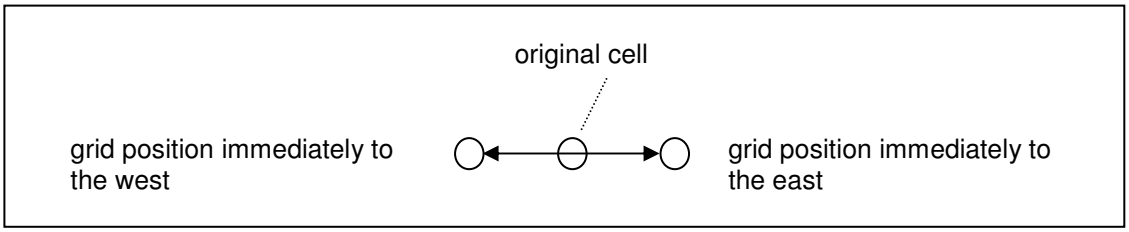


Figure 8-6 East, west growth rate, G_{EW}

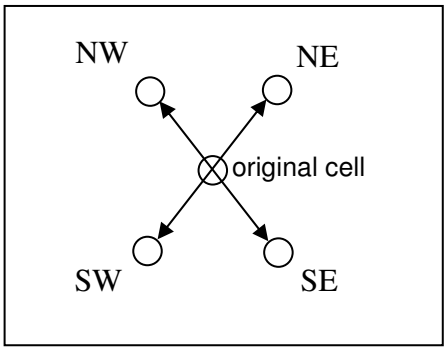


Figure 8-7 Growth Rate Along the Intercardinals

i.e. north-east, south-east, south-west and north-west directions, assigned a common growth rate G_{DIAG}

The three growth rate variables defined for each cell type are represented in Figure 8-5 to Figure 8-7, which depict respectively a north-south growth rate, G_{NS}, an east-west growth rate, G_{EW}, and a growth rate along the intercardinals (diagonals), $G_{DIAG.}$ A seed cell may thus replicate in two directions to the north and south as shown in Figure 8-5 according to the growth rate $G_{NS;}$ in two directions to the east and west as shown in Figure 8-6 according to the growth rate $G_{EW;}$ and in four diagonal directions, i.e. to the north-east, south-east, south-west and north-west, as shown in Figure 8-7 according to the growth rate $G_{DIAG.}$

The string *ch2* is used to store the values of G_{NS} for each cell type, the string *ch3* is used to store the values of G_{EW} for each cell type, and the string *ch4* is used to store the values of G_{DIAG} for each cell type.

A set of three trigger variables P_{NS}, P_{EW} and P_{DIAG} is also created for each cell type in and is stored in the memory to act as the trigger for cell reproduction. These are initially set to zero. Each cell type in the grid thus has three separate triggers, one for each of the growth rate axes. Each of the initial seed cell locations corresponding to grid location i_1 to i_M within the grid thus has three triggers P_{NS}, P_{EW} and P_{DIAG}. Once the value of one or more of the triggers has exceeded one, the cell can then form a new cell in the respective directions.

More especially, if the value of P_{NS} exceeds one, then two new cells can be formed in the grid locations to the north and south of the original cell, provided the grid locations are not already occupied. If the value of P_{EW} exceeds one, then a new cell can be formed in the grid locations immediately to the east and west of the original cell, two new cells in total, provided the grid locations are not already occupied by other cells. If the value of P_{DIAG} exceeds one, a new cell can be formed in the grid locations immediately to the north-east, north-west, south-west and south east, four in total, provided the grid locations are not already occupied by other cells. This process is shown in the flow diagram Figure 8-8.

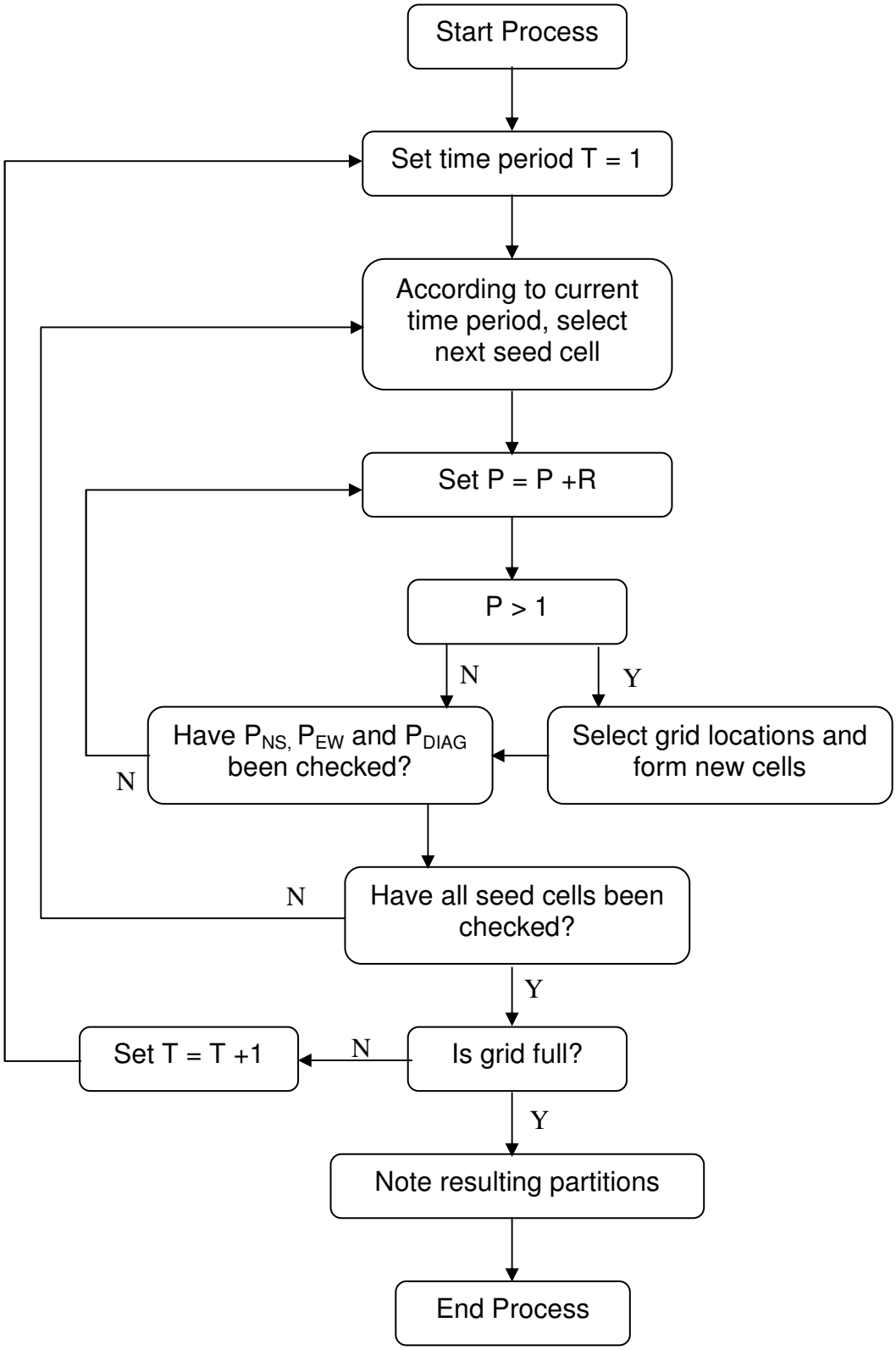


Figure 8-8 Cell Growth Process

Each new cell that is created is allocated to an available grid location adjacent to a respective existing cell or occupied grid location. The new cell's type will be that of its mother cell. When all the grid locations in the grid are occupied, partitions arise as defined by the boundaries of the different cell types as they have spread to fully populate the grid.

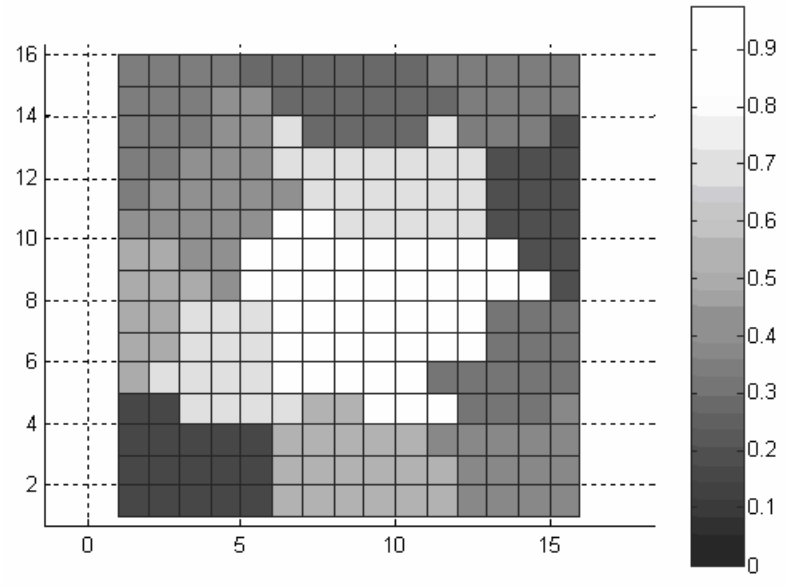


Figure 8-9 Example of 15 x 15 Element Partitioned Array

In total, the number of variables *N* required to control the partitioning process is given by Eq. 8-1:

$$N = M + 3M = 4M \qquad \textbf{Eq. 8-1}$$

where *M* is the number of subarrays.

8.4. Optimisation Example

What follows is an example of using the binary fission partitioning algorithm to determine a subarray partitioning configuration that offers an optimal compromise between mainlobe gain and maximum sidelobe level performance. Due to its success in the antenna test case problems, the ε-MOEA algorithm has been used.

This example assumes that the subarray partitions will be fixed during manufacture (typical of many existing active array antenna designs). The algorithm determines the subarray partitions and the amplitude weightings to apply uniformly to each subarray.

A simple square planar array containing 225 elements was designed to operate at 10GHz. The elements were arranged in uniform rows as per Figure 8-10.

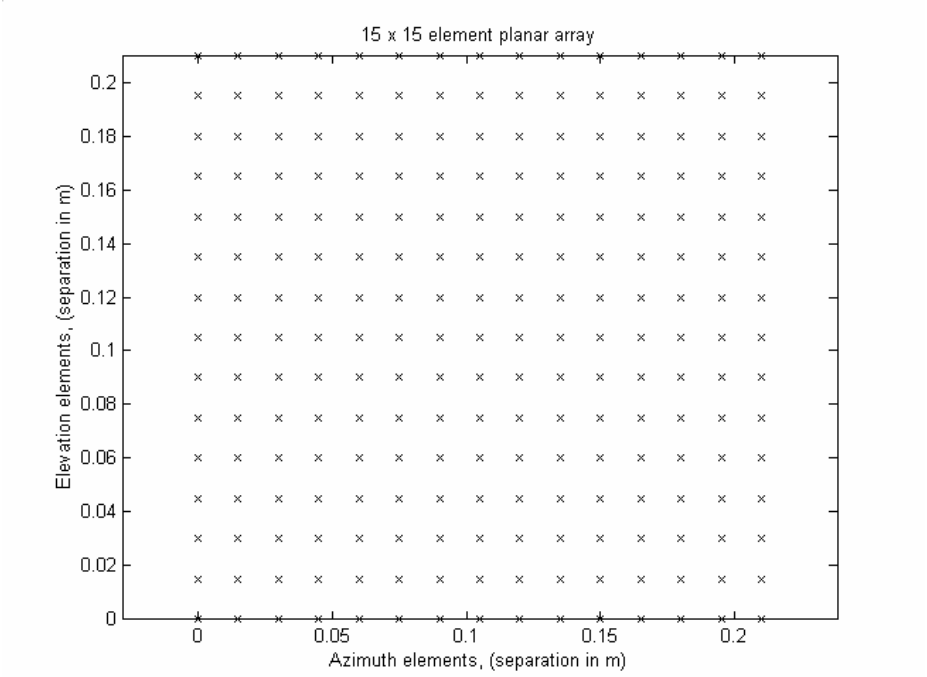


Figure 8-10 Test Case 15 x 15 Element Planar Array

Each element in the array was given a unique installed radiation pattern using the method described in Chapter 2, section 2.5. The unpartitioned array, when uniformly excited gave the radiation pattern shown below in Figure 8-11.

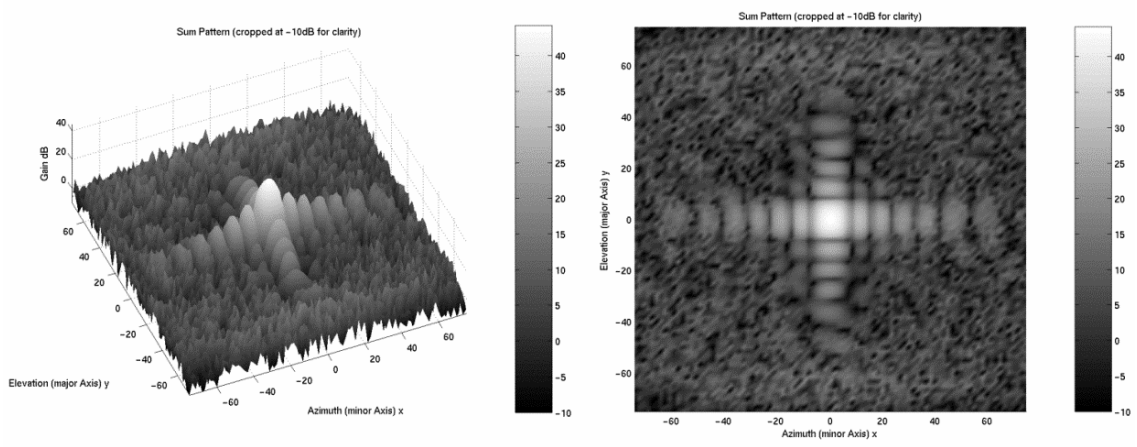


Figure 8-11 Pattern When Uniformly Excited

The unique element patterns make this optimisation example more realistic, and as they are included in the calculation of the radiation pattern, they influence the EA's search for good solutions.

The algorithm was stopped after 34,500 evaluations of the cost function. The archive solutions found are shown in Figure 8-12.

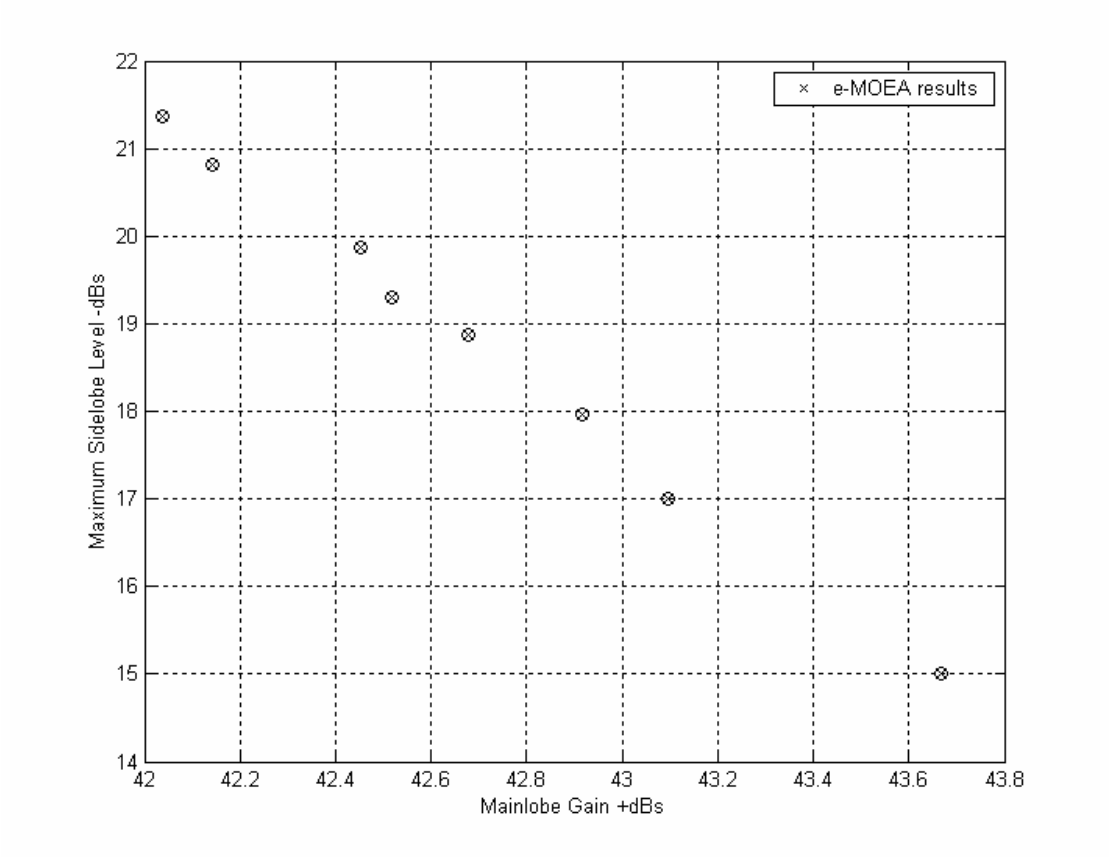


Figure 8-12 Archived Solutions

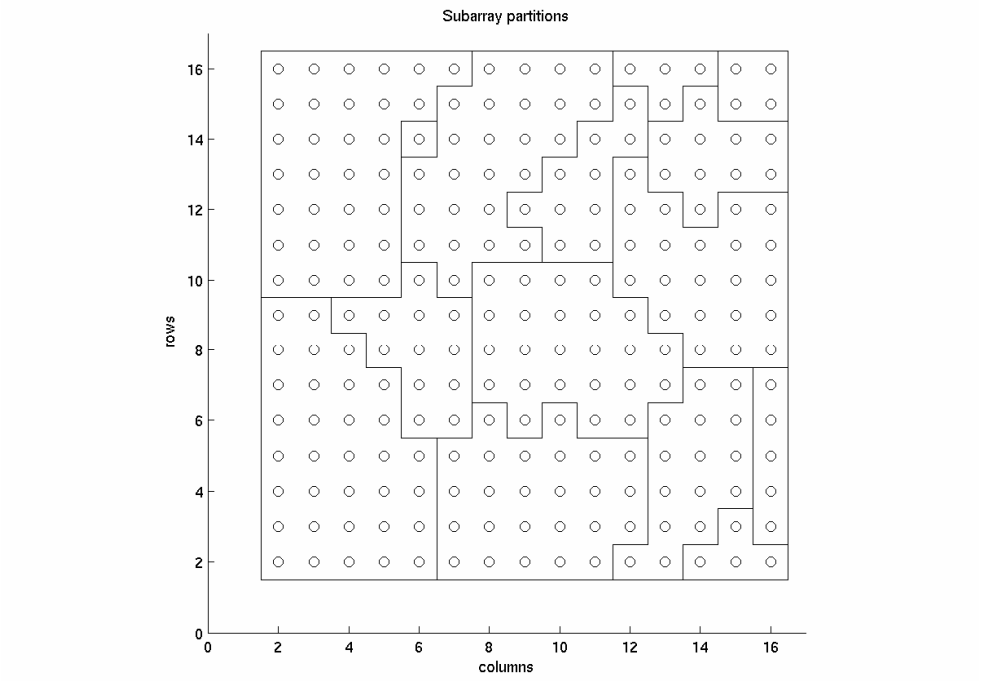


Figure 8-13 Optimised Partitions for Lowest Sidelobe Solution

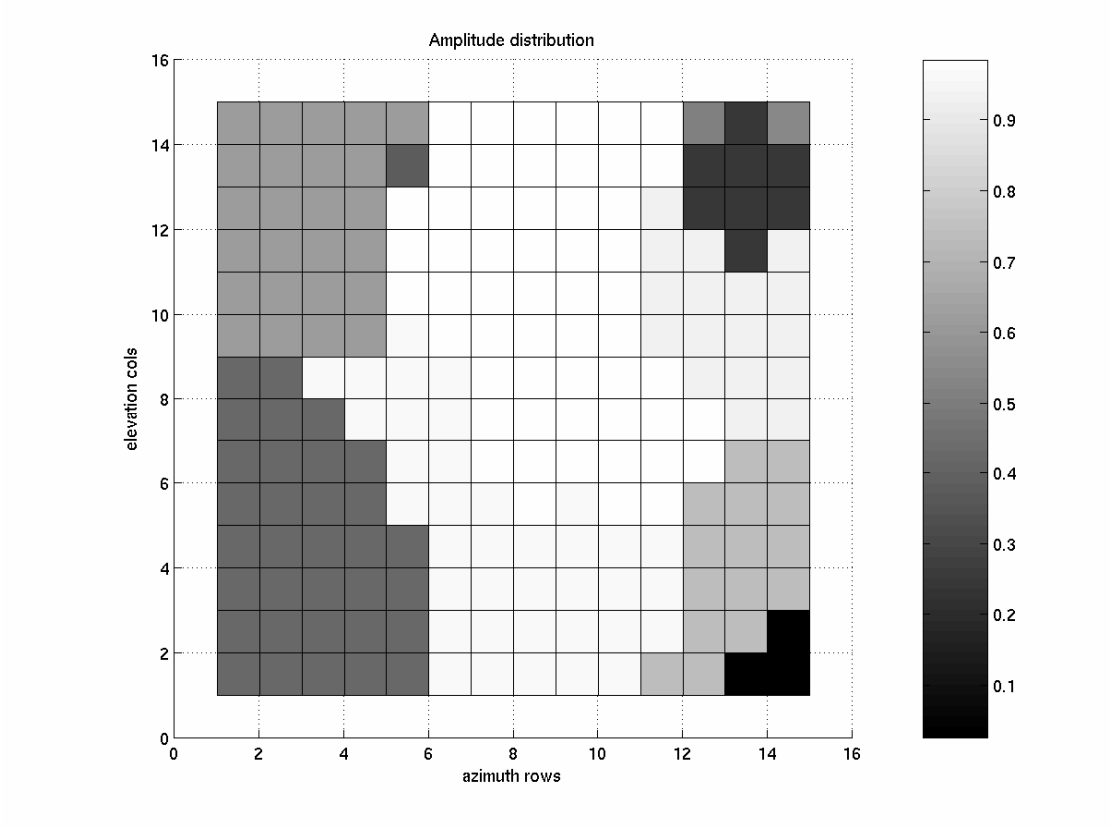


Figure 8-14 Optimised Amplitude Weightings for Lowest Sidelobe Solution

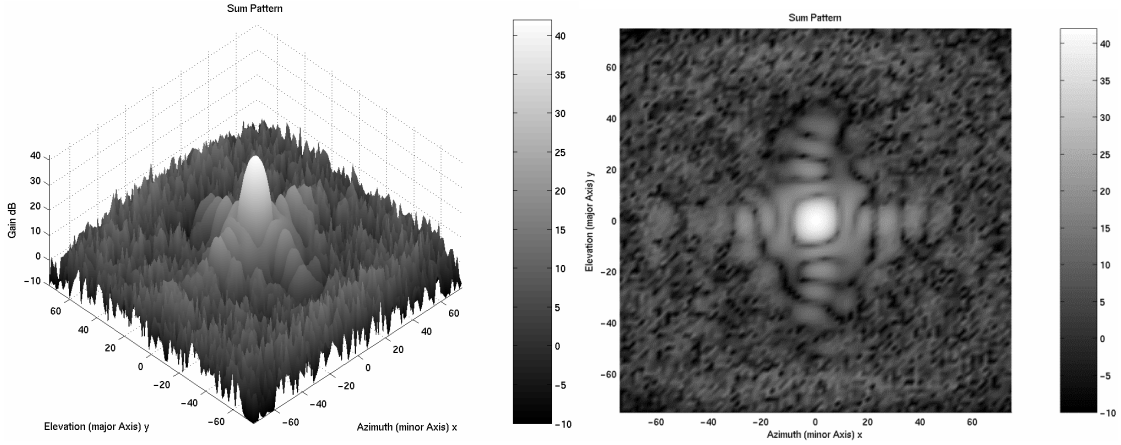


Figure 8-15 Complete Optimised Radiation Pattern

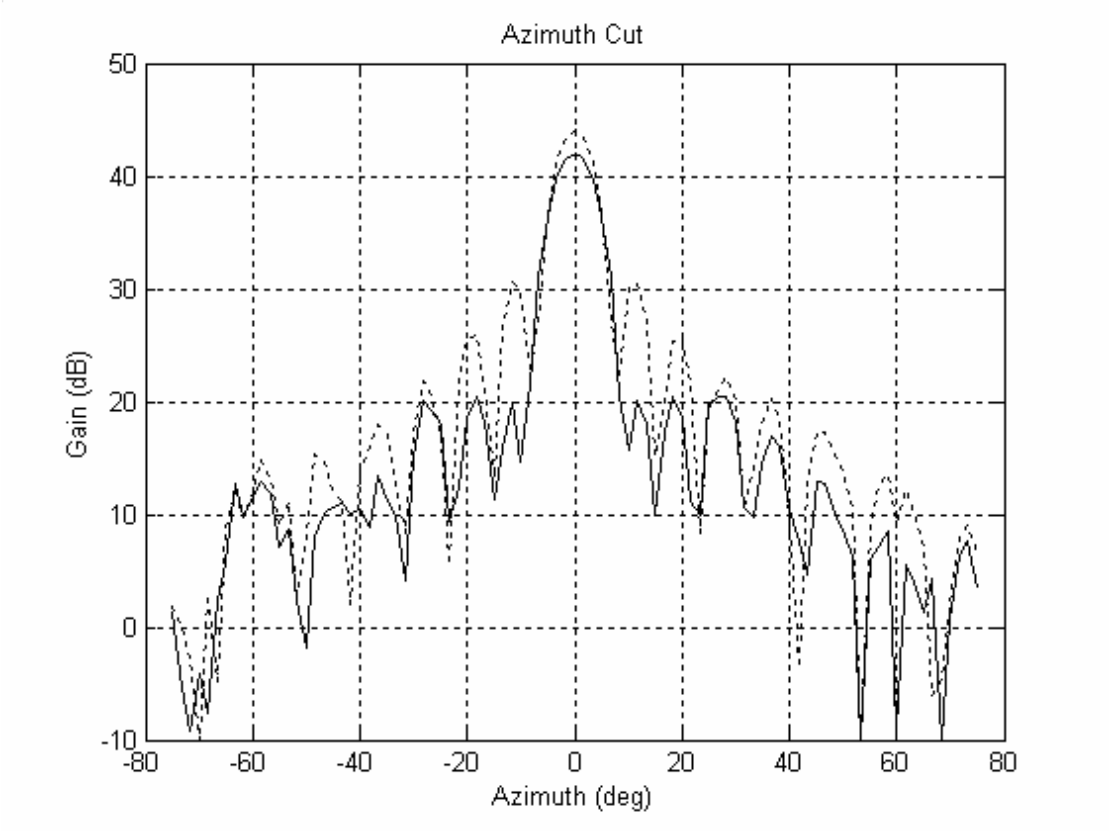


Figure 8-16 Comparison of Uniform Illumination and the Optimised Pattern

As the test case array is relatively small and the number of subarrays is few, very low sidelobe patterns that require detailed control of the excitation set cannot be achieved. Given these constraints, the optimisation results demonstrate what can be realistically achieved with this antenna configuration.

If lower sidelobe performance is required, the number of subarrays must be increased, or weighting must be applied at the element level as well as at the subarray level.

CHAPTER 9
Conformal Array Optimisation

9. Conformal Array Optimisation

9.1. Introduction

So far in the course of this work, the analysis and optimisation has been restricted to linear and planar arrays. In this section, we shall deal with optimisation of conformal arrays.

Planar arrays are usually limited to scan angles of about ± 60 degrees from boresight due to the characteristic broadening of the mainlobe and consequent reduction in gain. Scanning beyond these angular limits with planar arrays requires additional apertures and is not always practical, particularly in airborne applications where space is limited. The cost of the additional arrays would also tend to be prohibitive. An alternative approach is to move away from a planar array and instead, locate the radiating elements on a surface that conforms to a non-planar shape. Arrays of this type are collectively described as conformal arrays.

Conformal arrays are usually designed to conform to circular or elliptical arcs. Curved arrays have several advantages over linear or planar arrays:

- the field of regard can be improved particularly if the array conforms to an arc or spherical surface as the elements face in different directions,
- the space required for installation can be reduced – advantageous in airborne platforms,
- the increased field of regard could potentially enhance multifunction operation of conformal array antennas.

Unfortunately, there are several complications associated with the operation of conformal arrays. The hardware is expensive, more difficult to build and sidelobe performance comparable to planar or linear arrays is harder to achieve. To complicate matters, the elements that make up the array are all facing in different directions.

Consequently, the calculation of the array factor becomes more involved as detailed in Chapter two. Given that the element orientation can vary so much in conformal arrays, the polarisation must be carefully controlled either by constraining the geometry or perhaps by use of circularly polarised elements.

To steer the beam, it is necessary to alter the amplitude and phase of each element taking into account the installed element patterns. A conformal array with thousands of elements, covering a large field of regard and using different aperture distributions for every beam position will require considerable data storage, coupled with rapid data access. However, the advances in device and digital signal processing technology are reducing the costs of the necessary hardware - increasing the viability of using conformal arrays in real applications.

Ring arrays have the radiating elements around the circumference of a circle. Full circumferential scans may be accomplished by illuminating a sector of the arc and advancing the sector electronically to achieve the full 360 degree scan. It is unusual to see ring arrays in real applications, but arrays with elements arranged on an arc (typically less than 180 degrees) are used in some specialist communication systems [154].

Practical implementation problems of large conformal arrays are significant (such as cross-polarisation) and conformal radar antennas are unlikely to enter service for another ten or fifteen years. One of the problems is that radiation pattern control of conformal arrays is complicated due to additional geometry component. Excitation sets calculated for planar or linear arrays do not give the same performance when applied to curved antennas.

9.2. Optimisation of Conformal Arc Performance

The optimisation of excitation sets for conformal arc antennas is analogous to that for simple linear arrays. The array factor calculation changes, but the techniques developed earlier in this thesis (such as SWS in Chapter 5) for search of low sidelobe patterns can be applied here.

The radiation pattern for a simple conformal arc array [59] is given by

$$F(\phi) = \sum_{n=1}^{N} I_n \exp(j\alpha_n) \exp[jkR\cos(\phi - \phi_n)] FE(\phi - \phi_n) \qquad \textbf{Eq. 9-1}$$

Where

N = total number of elements,
I_n = the amplitude of element n,
α_n = the phase of element n,
k = 2 * pi / lambda,
R = radius of the arc,
ϕ_n = angular position of element n relative to the centre of the arc at $\phi = 0$.

The important element factor $FE(\phi)$ is also included to account for installed performance.

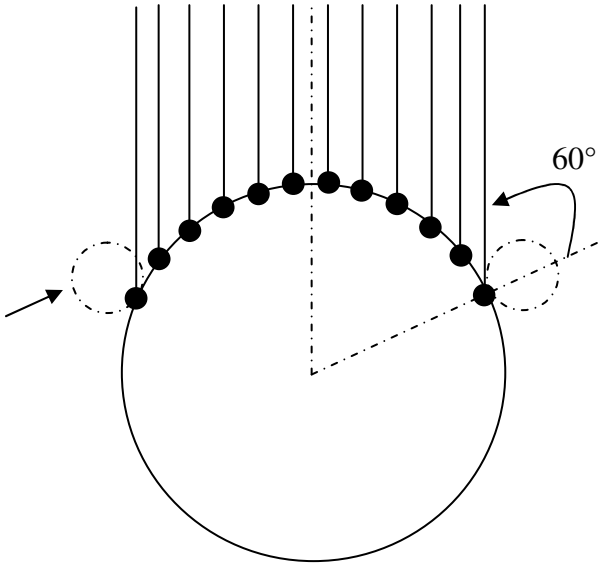


Figure 9-1 Conformal Arc Phased for Propagation at Φ = 0.

In order to steer the array, the phase of each element has to be calculated according to:

$$I_n = \left|I_n\right| \exp^{-jk[R\cos(\phi_0 - n\Delta\phi_k)]} \qquad \textbf{Eq. 9-2}$$

where Φ_0 = steering angle and $\Delta\ \Phi_k$ is the angular spacing between elements.

When considering the optimisation of conformal arrays, the definition of the radiation pattern objective measures is especially important and additional objectives are needed compared with some of the earlier linear and planar array optimisations. Initial attempts optimise conformal array patterns used just two objective measures to maximise the gain at the desired steering angle and minimise the sum of the pattern at all other sample points. These objectives were too vague to prevent high gain sidelobes from forming.

It was important to take into account particular characteristics that are common in conformal patterns. The far-out sidelobes are raised when compared with a linear array pattern, and these are often of greater magnitude than the sidelobes close to the mainlobe. Monitoring a single measure of maximum sidelobe level in the pattern will help to reduce the far out sidelobes, but will permit the first few sidelobes to vary in magnitude up to the maximum sidelobe level, compromising the pattern performance.

Use of an additional objective measure to monitor the sidelobe level within a short region (say +/- 40 degrees) of the mainlobe helps to guide the evolutionary process towards maintaining more desirable radiation patterns.

9.2.1. Conformal Arc Sidelobe Levels

All prior applications of genetic algorithms in conformal array pattern optimisation have used the simple single-objective GA. A multiple objective EA was applied for the first time to attempt to optimise sidelobe performance. The ε-MOEA [124] was used to

optimise the amplitude weightings for a 51 element conformal arc array. The arc array was designed to operate at 7 GHz. The radius of the arc was 45cm and the elements were arranged around the arc at half wavelength spacing. The elements were assigned unique element patterns generated using the procedure in Section 2.5. All the elements were arranged so that they radiated radially outwards.

Three objective values were used to measure the patterns produced; f_1 represented mainlobe gain, f_2 the maximum sidelobe found in the pattern (excluding the +/- 40 degree 'near-in' sidelobe region) and f_3 the maximum sidelobe experienced in the +/- 40 'near-in' region (as defined in Section 2.4). The chromosome contained the entire set of amplitude weights (fifty-one in total) to be applied at the element level.

The dotted line in Figure 9-3 shows the radiation pattern of the conformal arc array when illuminated with a uniform excitation. The mainlobe experiences its maximum possible gain of 30.59 dB, the maximum value of the near sidelobes is -10.28dB (relative to the mainlobe) and the maximum sidelobe found in elsewhere in the pattern is -17.79 dB (relative to the mainlobe). By contrast, the chosen optimised solution (marked as Solution A) has 27.48dB mainlobe gain, -19.93dB near-in sidelobe level and -24.56 dB far-out level.

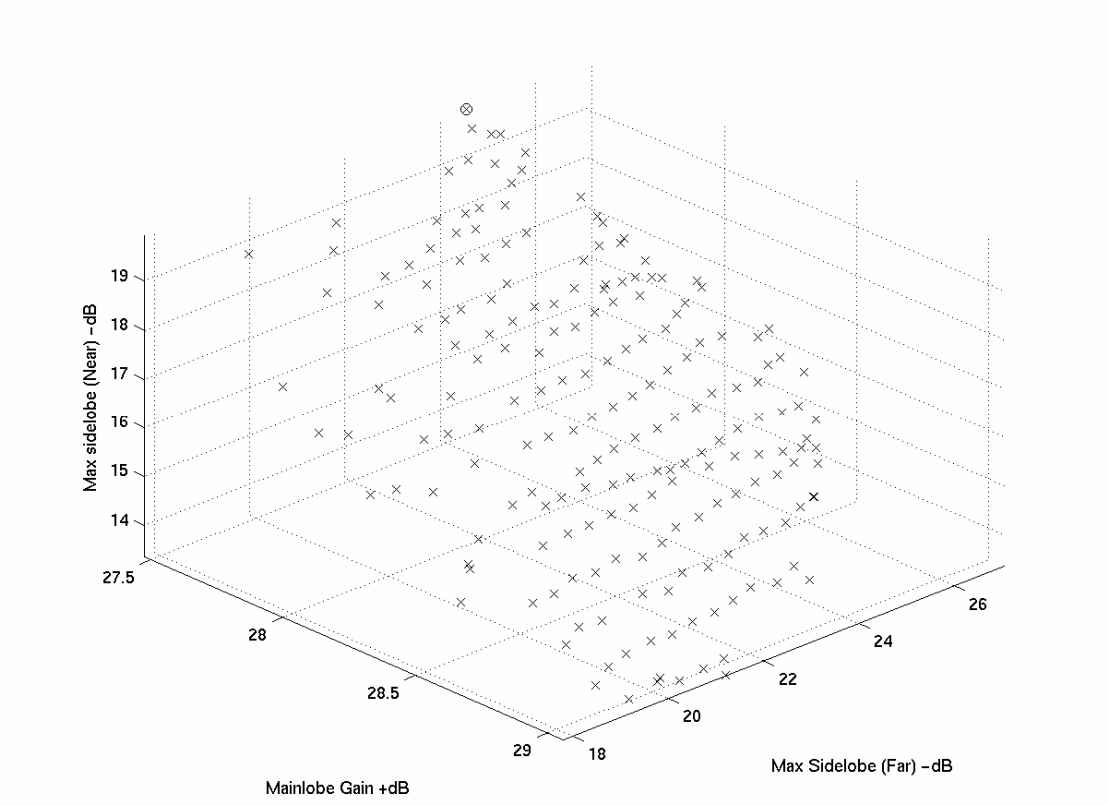


Figure 9-2 Final Pareto Archived Solutions Found Using ε-MOEA

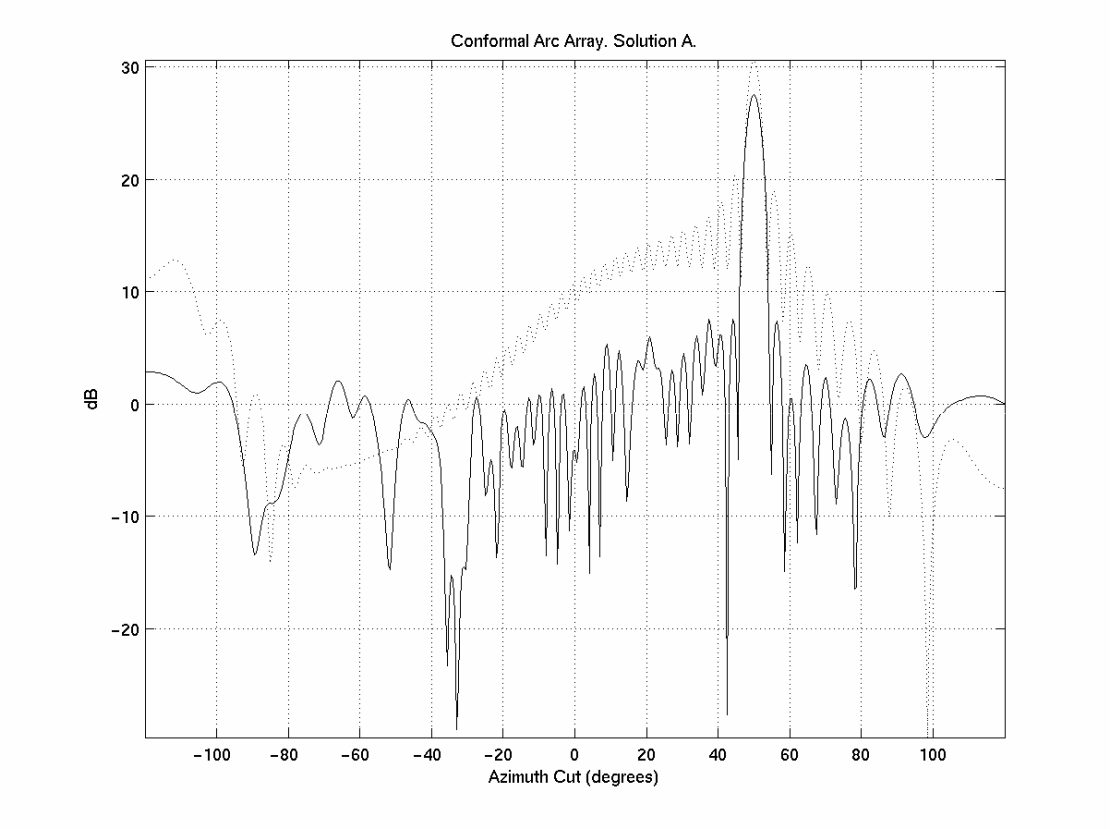


Figure 9-3 Example 'Solution A' from the Pareto Archived Solution Set

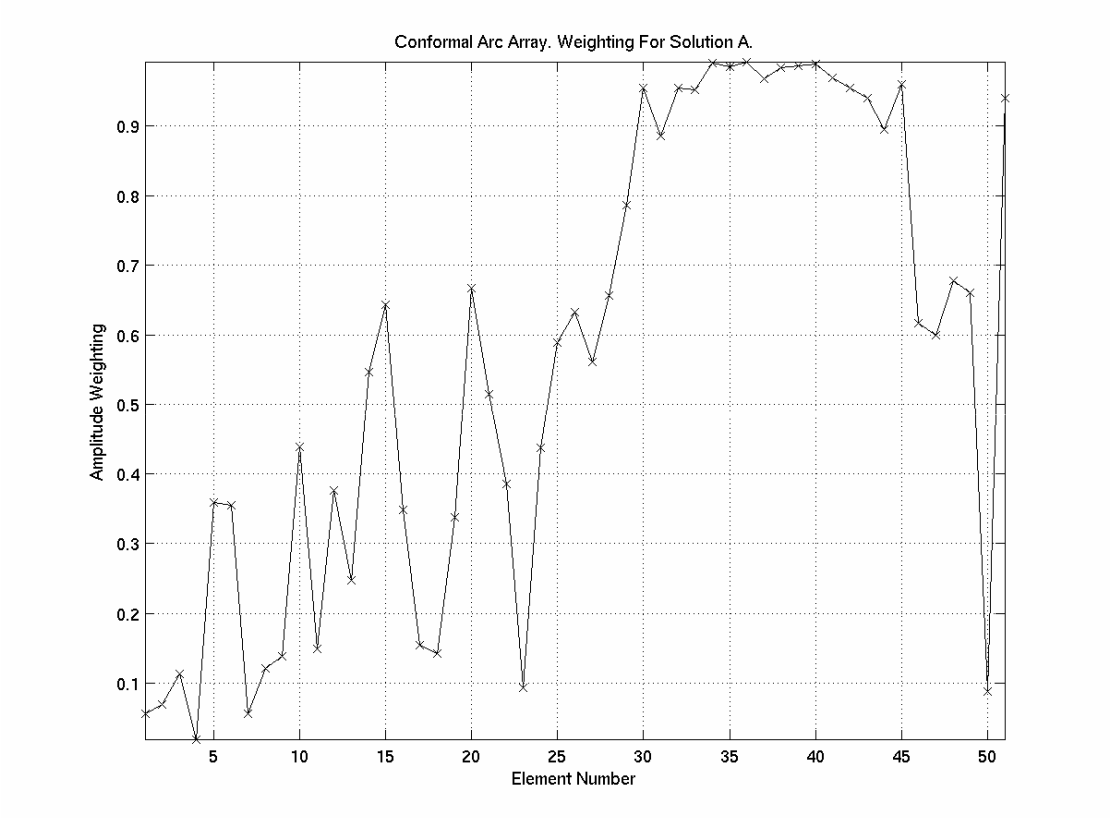


Figure 9-4 Optimised Amplitude Weightings for Solution A

While the EA found a good spread of solutions, and the performance of solution A was a major improvement over the uniform illumination case, the method suffers from the same problems experienced when optimising linear arrays - that of an irregularly shaped amplitude taper. The performance of solution A is typical of much prior work in this area reported in Chapter 1. The final non-dominated set contained 196 unique solutions.

In an attempt to improve optimiser performance, the optimisation run was repeated but this time using the SWS encoding method. There was a noticeable improvement in the results set which confirmed that SWS is an efficient genotypic encoding method for use on conformal arrays. Figure 9-5 shows the improved Pareto archived solution set that contains 435 solutions.

A solution was chosen from a similar region of the Pareto set (labelled B) and its performance is detailed in Figure 9-6 and Figure 9-7. Solution B has a mainlobe gain of 25.64dB, a maximum far region sidelobe level of -38.49dB and a near region sidelobe level of - 21.11dB.

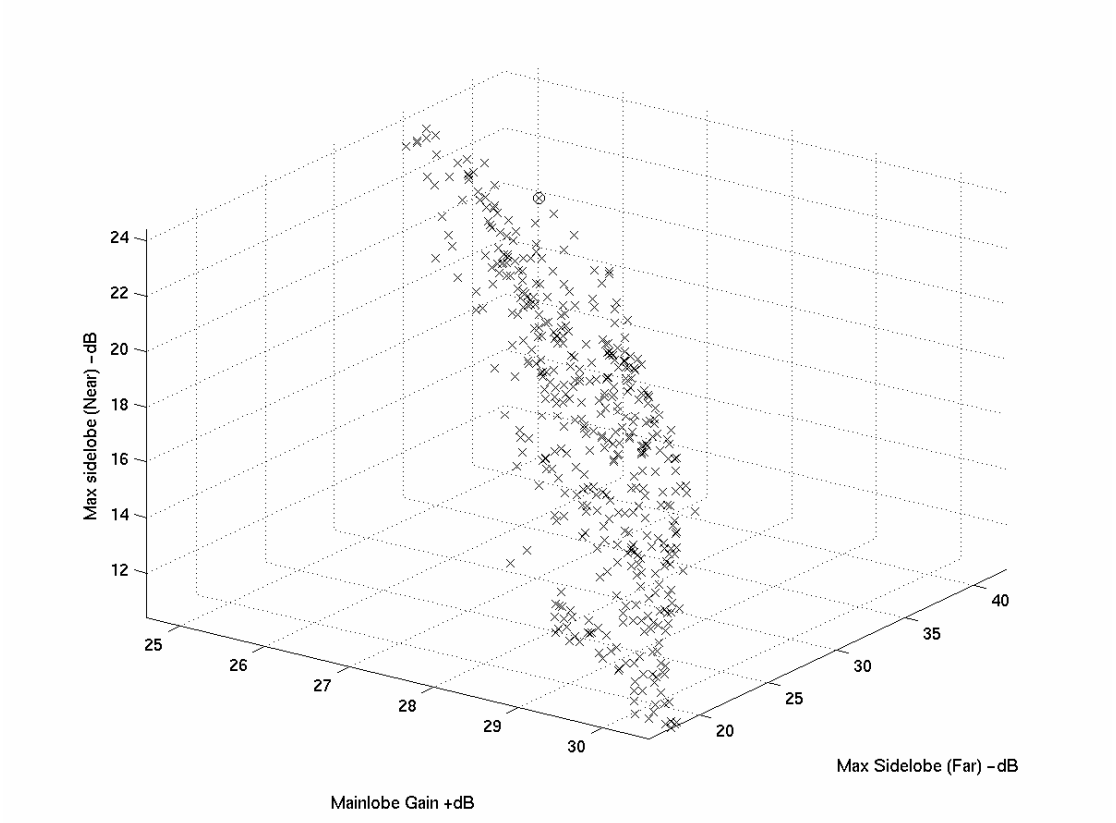


Figure 9-5 Pareto Archived Solutions Found Using ε-MOEA and SWS

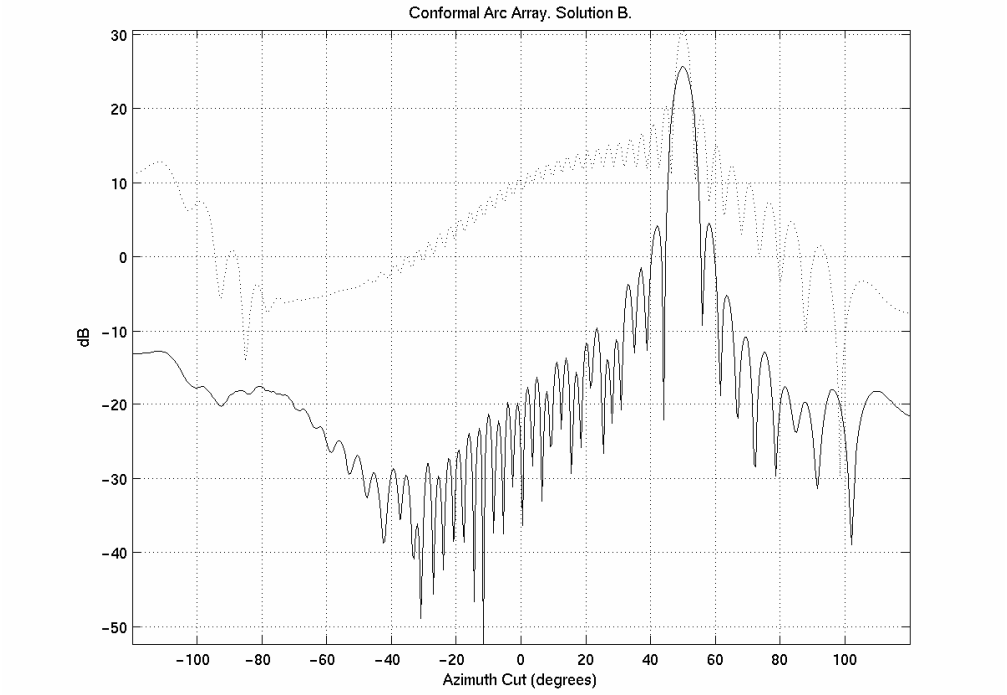


Figure 9-6 Example 'Solution B' from the Pareto Archived Solution Set

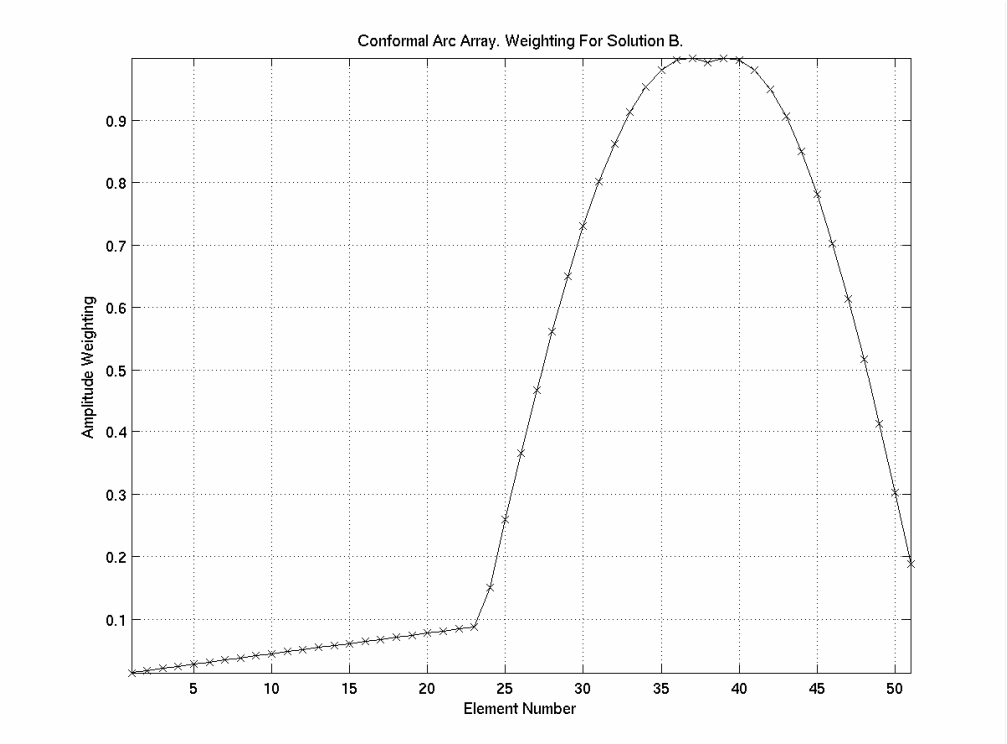


Figure 9-7 Optimised Amplitude Weightings for Solution B

The regularity of the amplitude taper appears to be key to obtaining good performance when optimising sidelobe level. It was discussed in Chapter 5 that other researchers have tried to encourage a regular shaped taper by modifying the cost functions to include an objective that minimises the difference between successive amplitude

weights. Complicating the cost function can be counter-productive as it can slow down convergence. The results above demonstrate that there are occasions where it is better to use the genotype to promote occurrence of desired characteristics, rather than attempt to achieve them via the cost function.

9.2.2. Difference Pattern Optimisation

In order to track the position of the satellite, the array must be capable of producing difference patterns. A second example was formulated to demonstrate that an EA could be used for this purpose.

The method presented below for the optimisation of conformal array difference patterns is similar to the earlier example for a linear array. A single cut of the radiation pattern is optimised. A single cut is sufficient for demonstration of the technique and speeds-up run times considerably.

The first step of the EA partitioned one half of the conformal array's elements into subarrays and excitations were applied to each subarray using the simple divisors method described in Chapter 8. A reverse mirror image of excitations (where the phases are inverted) was then calculated. This image formed the second half of the array's excitations and also defined the remainder of the partitioning.

The EA used in this example was the SPEA2 algorithm four objective values were used. This analysis was for propagation along bore-sight only. For the purpose of analysis of the radiation patterns, each pattern produced was divided in two, with each half being analysed separately. The division made it easier to measure multiple mainlobe formation using existing analysis code.

The left hand side of the pattern (-120 deg <= phi <= -1 deg) was analysed in order to determine the maximum gain, the maximum sidelobe level and location of the mainlobe produced. This process was repeated for the right hand side (1deg <= phi <= 120 deg).

Averages or sums of measurements were taken where necessary in order to reduce the total number of objectives to a reasonable number e.g.

maximum sidelobe gain = (max sidelobe gain LHS + max sidelobe gain RHS) / 2.

Eq. 9-3

The combination of measurements resulted in four different objective measures being required to define a pattern:

f_1 = The gain of the mainlobes
f_2 = Maximum sidelobe level
f_3 = Depth of the centre null
f_4 = Location of the mainlobes

As there were 51 elements in the array, the total number of subarrays was set at a maximum of ten. To speed up the computation time, a cut along the azimuth axis was

optimised, rather than the entire pattern. As there are a single row of elements in this array, the results should hold true for three-dimensional radiation pattern optimisation.

The basic GA parameters were set as: Population size = 150, Archive size = 100, P_{cross} = 90% and P_{mut} = 5%. Uniform crossover operators where used along with single-bit mutation operators.

As discussed in Chapter 6, the higher number of objectives included, raised the numbers of non-dominated solutions existing at each generation. This problem was somewhat mitigated by using a large population size. During the fifth generation and thereafter, the truncation operator was required to reduce the archive size.

The final archive contained a diverse set of solutions, and Figure 9-8 shows one chosen to have the lowest sidelobe level performance.

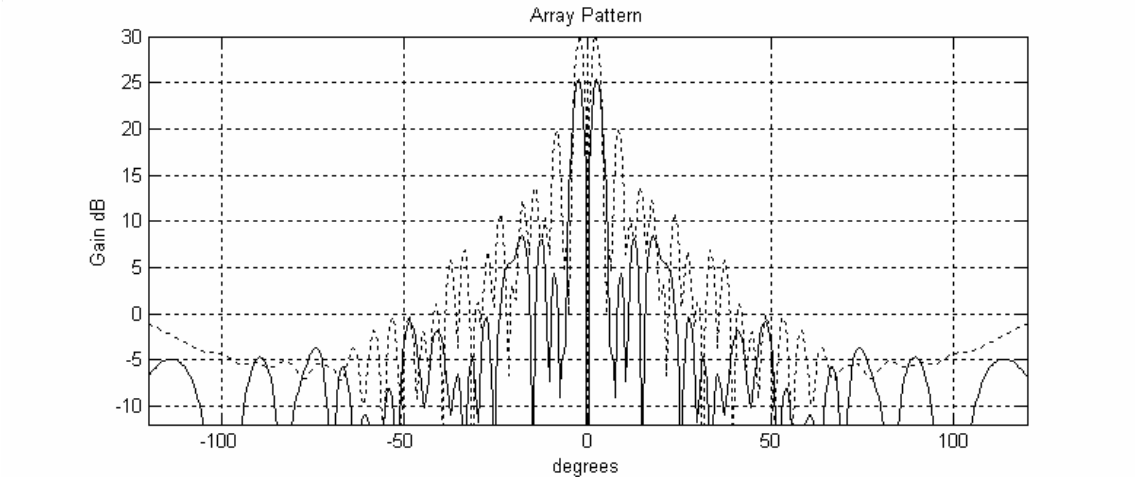


Figure 9-8 SPEA2 Optimised Solution for a Conformal Arc Array

In Figure 9-8, the dotted line shows the array with uniform amplitude weighting, while the solid line shows the optimised solution. The uniform pattern has mainlobe gain of 30.06dB, a relative sidelobe level of -10.23dBs and a centre null depth of -149.38dB. The chosen optimised solution has a mainlobe gain of 25.27dB, a relative sidelobe level of -16.84dB and a centre null depth of -164.9dB. The excitation set for this solution is given in Figure 9-9 . The excitation set in Figure 9-9 can be achieved by using a set of positive amplitude weights and by adding 180 degrees to the phase shift applied to the LHS elements.

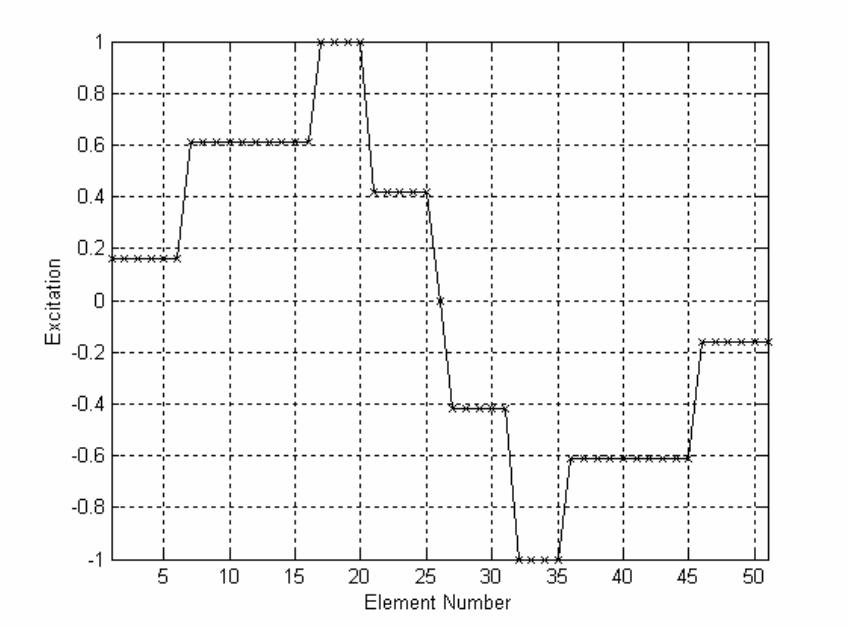


Figure 9-9 SPEA2 Optimised Excitation Set

Initially, one hundred generations were completed and reviewed, but there proved to be little change after a further one hundred generations had been completed.

In multiple objective problems it can be difficult to monitor convergence graphically, but there was sufficient similarity between archive sets to give confidence that the set had converged.

9.2.3. Multiobjective Optimisation of a 15 x 15 Element 3D Array

This example uses the binary fission partitioning algorithm and ε-MOEA to optimise radiation pattern performance of a 15 x 15 element conformal array antenna.

Although all applications of the binary fission partitioning algorithm have been on planar arrays, the partitioning can be applied to curved arrays as it is an index based (rather than geometry based) system. An example of the partitioning results obtained when the binary fission algorithm was applied to a conformal array is shown in Figure 9-10.

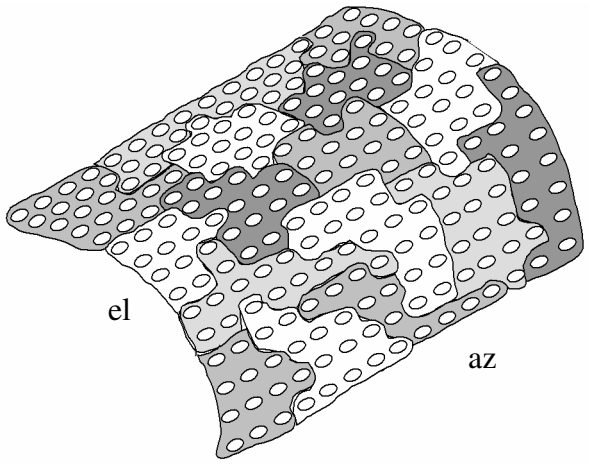


Figure 9-10 Example of Conformal Partitioning Using Binary Fission Algorithm

The geometry of the array is shown in Figure 9-11.

In order to determine the success of a solution (defined by one set of chromosomes), it is necessary to model the far field pattern of the antenna. The far field radiation pattern of the conformal array was calculated using:

$$F(\theta,\phi) = \sum_{n=1}^{N} a_n \exp(j\alpha_n) \exp[jk(x_n \sin\theta\cos\phi + y_n \sin\theta\sin\phi + z_n \cos\theta)].FE_n(\theta_i,\phi_i) \quad \text{(Eq. 1)}$$

The steering phase required to steer the beam to θ_o, ϕ_o was calculated using:

$$\alpha_n = -k[x_n \sin(\theta_o - \theta_n)\cos(\phi_o - \phi_n) + y_n \sin(\theta_o - \theta_n)\sin(\phi_o - \phi_n) + z_n \cos(\theta_o - \theta_n)] \quad \text{(Eq. 2)}$$

The pattern has been optimised with respect to the following objectives:

- f_1 = maximise mainlobe gain,
- f_2 = minimise the difference between the azimuth and elevation beamwidth,
- f_3 = minimise sidelobe level in the pattern within 40 degrees of the mainlobe in both azimuth and elevation,
- f_4 = minimise sidelobe level in all other regions of the pattern.

Calculating the objective f_2 required measurement of the 3dB beamwidth of the mainlobe. Measurements were taken in both azimuth and elevation and f_2 was set equal to the magnitude of the difference in the measurements (and then negated so that it could be maximised).

The two separate sidelobe level objectives were necessary due to the higher magnitude sidelobes that are experienced in this particular array design. If just one maximum sidelobe level objective were used, the sidelobes close to the mainlobe (the near-in sidelobes) would never influence the search outcome, and in theory could rise to be as high as the far-out sidelobes. Setting a second measure of sidelobe level that excludes the farthest out sidelobes ensures that patterns are evolved with low sidelobes in the region of the mainlobe.

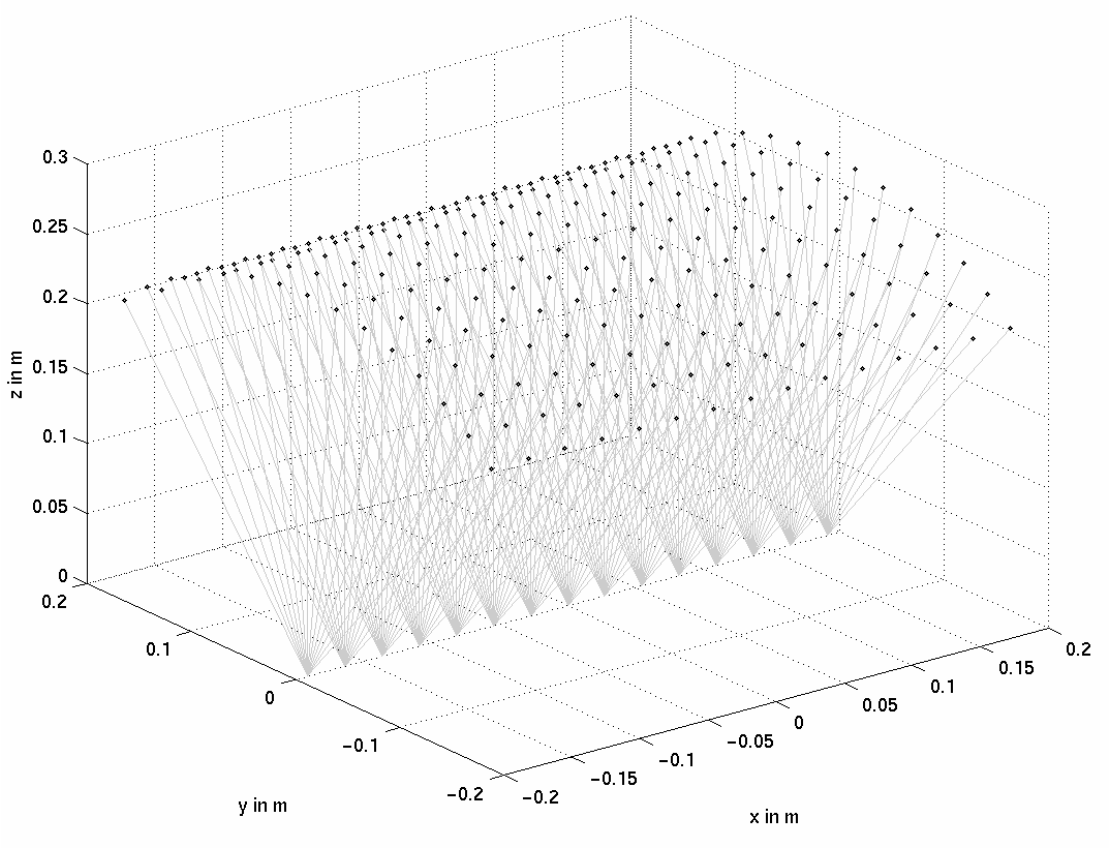


Figure 9-11 15 x 15 Element Conformal Array Geometry

The ε-MOEA was set up with an initial population size of 150 individuals. The population size was large enough to ensure that some dominated solutions existed to guide convergence in the algorithm. The algorithm was stopped after 22500 trial patterns had been evaluated - the equivalent of allowing a generation-based EA to run for 150 generations.

Five chromosomes were used, and each contained 15 variables. The chromosomes contained (1) seed cell locations, (2) north-south growth rates, (3) east-west growth rates, (4) diagonal growth rates and (5) subarray amplitude weights.

The amplitude weights were real valued, but were quantised to 5-bits in the antenna modelling function. Similarly, although phase was not optimised, phase values were quantised to 3-bits. Simulated element patterns were used to make the optimisation more representative of a real array.

The final archive solution set was found by taking a non-dominated sort of the solutions occupying each hyperbox. At the end of the algorithm, 3722 hyperboxes contained solutions, and these resulted in 504 unique and diverse non-dominated solutions that represent the best approximation of the Pareto solution set found.

The final set was sorted according to sidelobe level and one solution was chosen for illustration below. This solution had the lowest near-in sidelobe level, and represented one 'corner' of the Pareto archive.

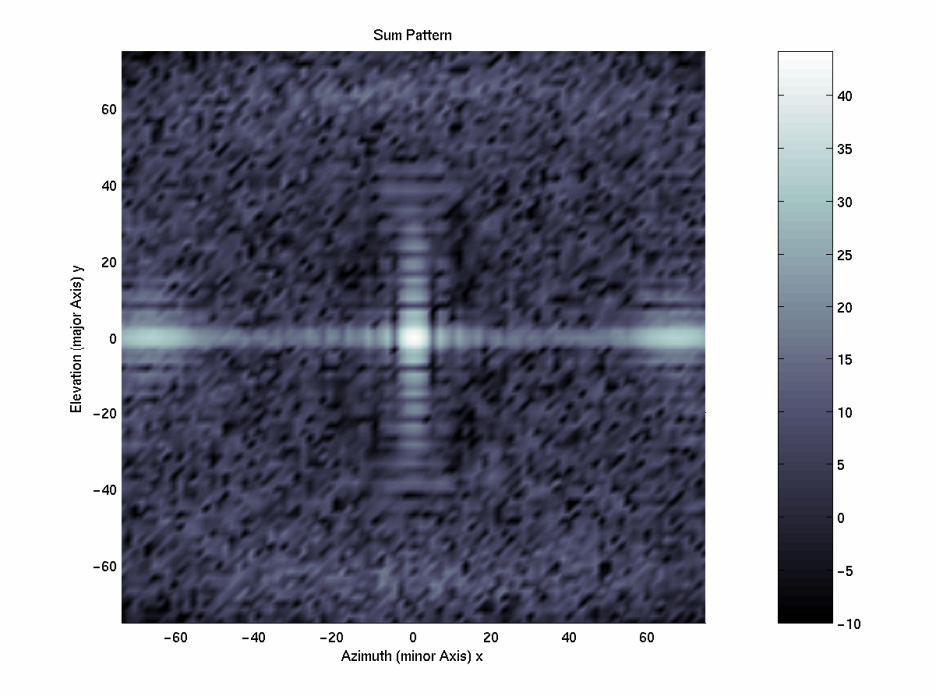


Figure 9-12 Radiation Pattern With Uniform Illumination

Figure 9-12 shows the radiation pattern for the array when a uniform amplitude taper (i.e. all weights set to one) was applied. The mainlobe gain is 44.20dB, the difference between azimuth and elevation beamwidths is -0.72deg, the maximum near-in sidelobe level is -14.12dB and the maximum far-out sidelobe level -11.33dB (all sidelobe levels quoted are relative to the mainlobe gain).

Figure 9-13 shows the example optimised solution. By comparison the pattern achieved mainlobe gain of 40.32dB, beamwidth error of -0.6deg a maximum near-in sidelobe level of -21.50dB and a far-out sidelobe level -11.88dB.

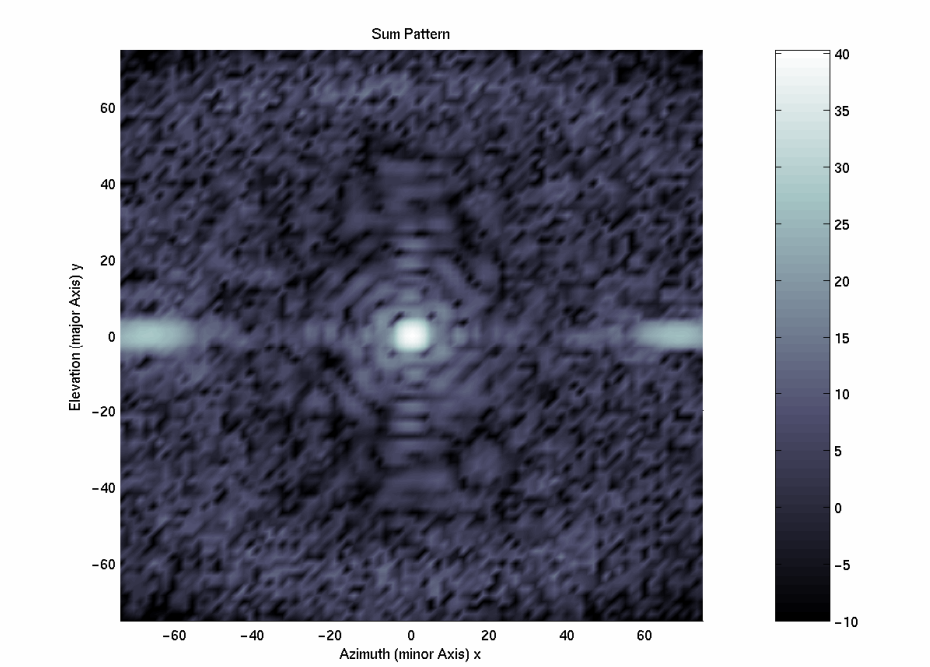


Figure 9-13 Radiation Pattern With Optimised Partitions and Excitations

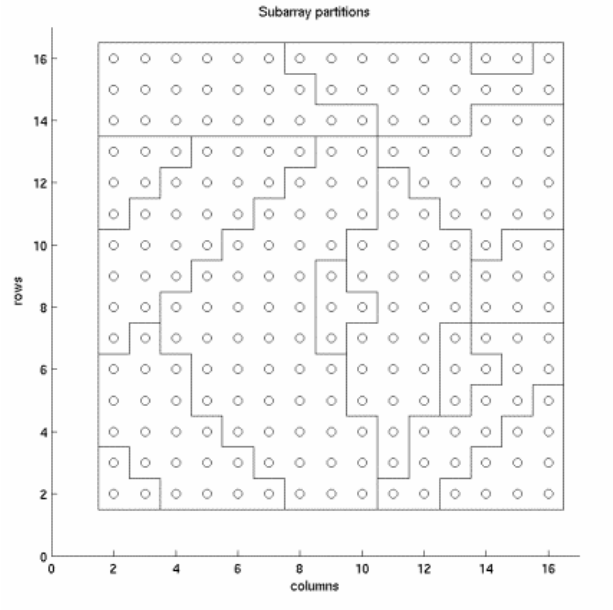


Figure 9-14 Optimised Partitions.

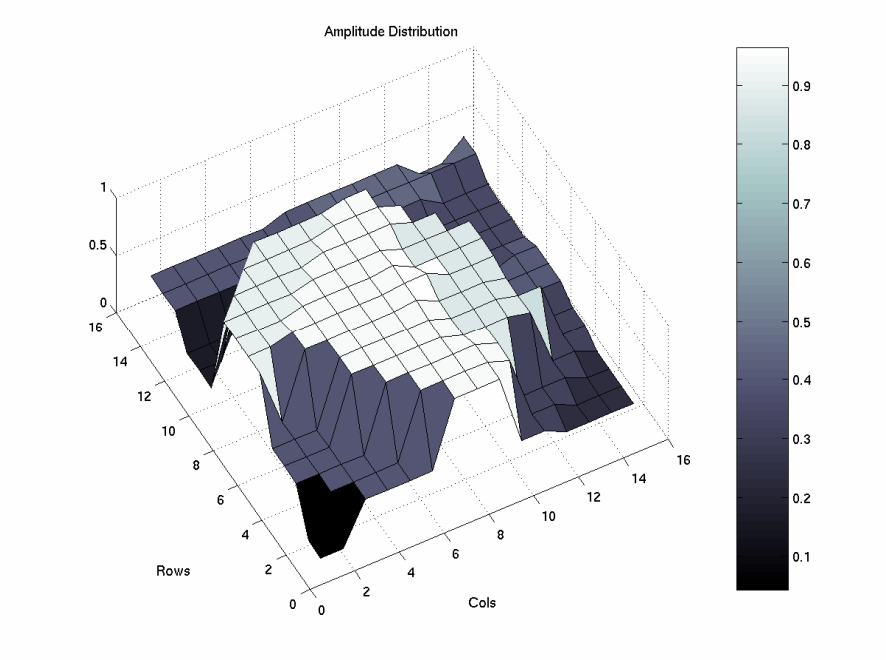


Figure 9-15 Optimised Amplitude Distribution

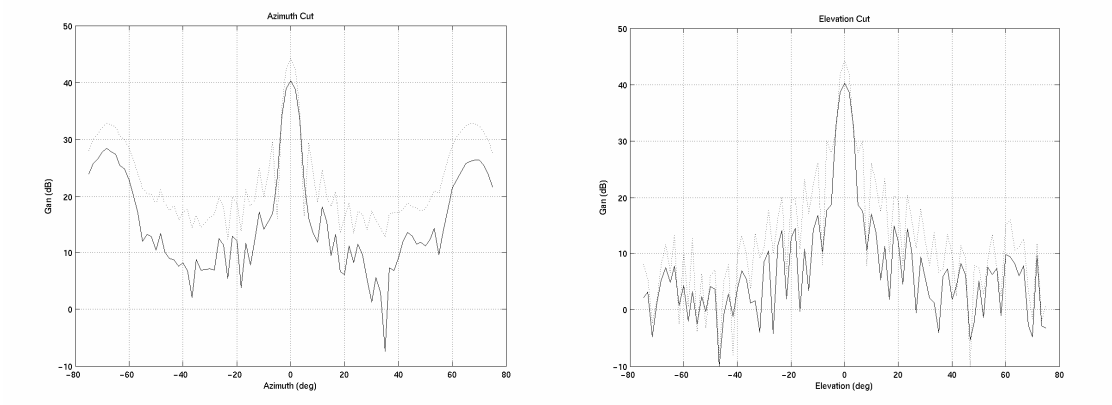


Figure 9-16 Azimuth and Elevation Cuts of Optimised Pattern.

It is interesting to note that the final excitation set has attempted to achieve a Taylor-like bell shaped distribution by setting the amplitude weights to higher values in the centre of the array. This characteristic suggests that a good compromise between sum and difference pattern performance can be achieved using a fixed set of partitions.

CHAPTER 10
Case Studies

10. Case Studies

10.1. Introduction

The purpose of this chapter is to demonstrate the versatility of multiobjective evolutionary algorithm optimisation in the optimisation of antenna radiation pattern performance. A number of examples have been presented in earlier chapters, but the case studies shown below detail how EAs can find well performing practical solutions for a variety of different RF functions.

10.2. Optimal Compromise Between Sum and Multiple Beam Patterns.

10.2.1. Problem Statement

In a fully digitised array (i.e. an array with amplitude and phase control at the element level and a separate receiver for each element), it is relatively easy to use a classically derived amplitude taper such as a Taylor to promote low sidelobe patterns, or a Bayliss function to produce multiple beam patterns. These arrays have the advantage of being able to dynamically change subarray boundaries according to function, and to apply different amplitude tapers at the element level, within each subarray.

Fully digitised arrays are expensive and are currently developmental technology and it is more common to see a smaller number of receiver channels that mandate fixed subarray partitions.

In this case study we assume that an active array antenna containing 1600 elements is to be divided into 64 subarrays and used in an airborne fire control radar system. In this capacity, the radar's operating modes require the array to produce low sidelobe sum patterns, and low-sidelobe multiple beam patterns. Phase control is provided at the element level in order to steer accurately the radiated beams. Amplitude control is only available at the subarray level.

The array is currently in the design stage and a single subarray partitioning configuration needs to be found that will optimise and promote the formation of the two stated types of patterns. Note: the multiple beam patterns could be simplified to contain just two lobes for difference pattern functions. This exercise demonstrates that the developed methods can produce good results on more complicated patterns and so chould be readily applied to a sum/difference pattern compromise investigation.

10.2.2. Solution

One way of solving this problem is attempt to find a set of subarray partitions that when weighted with two different sets of amplitude weights, would reproduce Taylor weighting approximations for the sum pattern and Bayliss weighting approximations for the multiple beam pattern. Figure 10-1 shows a two-dimensional Taylor amplitude weighting for the 1600 element array, and the contrasting Bayliss weighting. Not surprisingly, the Bayliss resembles, four 'smaller' Taylor distributions, one for each quarter of the array. The Bayliss provides four closely spaced mainlobes that could be used for various search functions or data link tasks.

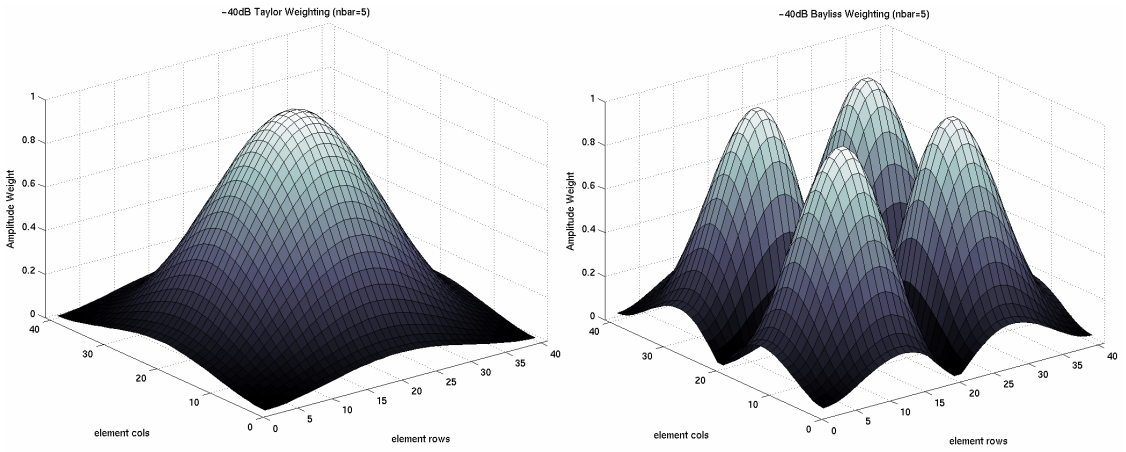


Figure 10-1 Taylor Taper (left), Bayliss Taper (right).

One property that both the Taylor and Bayliss weightings share is that they are symmetrical. If the taper is divided equally into four squares, the portion of the taper for a single square can be rotated or flipped and applied to the remaining three quarters of the array to reproduce the original taper.

This property reduces the amount of unique subarray partitions needed, as once found, they too can be rotated and the same designs reused in each quarter. Therefore the optimisation problem reduces to one of searching for 16 subarrays that completely partition one quarter of the array. This approach has practical benefits as reducing the number of unique subarray designs required helps to reduce the component cost of the system.

Once a trial set of subarray partitions have been produced, the next issue is how best to approximate the required taper using the subarray, given that weighting is applied at the

subarray, rather than at the element level. The single weighting has to someway represent the Taylor or Bayliss taper as applied at the element level. One method is to examine the Taylor weights that would applied to each subarray's member elements if they could be weighted at the element level. Collectively these weight values could then be averaged, or set to the minimum or maximum of the range.

Initial attempts to optimise the weights were unsuccessful as the variation in subarray shapes meant that a fixed method for weighting the subarray was too inflexible and the final results did not represent the taper.

The solution to this problem was to find both the partitions and the subarray weights using a multiobjective EA. Prior work in the course of this research has shown that if the optimised taper is very similar to a known well performing taper, the better the radiation pattern performance will be. Therefore, it follows that most of this optimisation exercise can be completed without any antenna models being used in the cost function. The cost function required is instead, a measure of correlation between the optimised taper achieved by weighting the subarrays and the required Taylor and Bayliss tapers. Any resultant optimised tapers must be ultimately tested using the antenna model, as it has been shown in earlier chapters that a relatively small number of incorrectly weighted elements can raise sidelobe levels significantly, and there can be no guarantee any tapers found to have a high correlation with the Taylor or Bayliss will necessary offer similar levels of radiation pattern performance.

A random trial solution was generated using the binary fission algorithm, first introduced in Chapter 8. The binary fission algorithm was used to partition one quarter of the array (i.e. 400 elements divided into 16 subarrays). Two different sets of weights were generated for the quarter, and then it was then rotated and flipped to partition the remaining three-quarters of the array. The result was two different amplitude tapers based on one set of partitions.

The first of these amplitude tapers was compared with the true Taylor amplitude taper by means of a correlation function. The correlation coefficient r was calculated using Eq. 10-1:

$$r = \frac{\sum_m \sum_n (A_{mn} - \overline{A})(B_{mn} - \overline{B})}{\sqrt{\sum_m \sum_n (A_{mn} - \overline{A})^2 \sum_m \sum_n (B_{mn} - \overline{B})^2}} \qquad \text{Eq. 10-1}$$

where A_{mn} is the Taylor weight for the element located at the mth row and nth column of the array, B_{mn} is the weighting of the current trial solution, $\overline{A} = \text{mean}(A)$ and $\overline{B} = \text{mean}(B)$.

Similarly, the second taper was compared with the true Bayliss taper to give a second correlation coefficient. The aim of the optimisation exercise is to maximise the two values of correlation by optimising the partition boundaries and the weights. The

algorithm is similar in many respects to a target recognition algorithm used in the image-processing domain.

The ε-MOEA was used due to the simplicity of the cost function in this problem, and the algorithm's successful use with the binary fission partitioning algorithm. The population size was set at 200, probability of crossover at 90% and mutation at 5%. The total number of optimisation variables was 96, comprising of 64 partitioning variables and 32 subarray weight variables.

Achieving perfect correlation (where $r = 1$) is impossible due to the uniform weighting of each subarray, so instead we aim to maximise the two values of r. Similarly, achieving the same levels of radiation pattern performance when using the optimised tapers (-40dB sidelobe performance) is unlikely for the same reason. The intention is to find a good compromise, given that the array must be weighted at the subarray level.

The algorithm was set to evolve for 100000 generations and the total computation time was around ten minutes due to the simplicity of the cost function. The resulting Pareto optimal set contained fifteen solutions with correlation values as shown in Table 10-1 and Figure 10-2

Solution	Taylor Correlation Coefficient	Bayliss Correlation Coefficient
1	0.30	0.87
2	0.92	0.76
3	0.89	0.80
4	0.92	0.77
5	0.21	0.87
6	0.46	0.87
7	0.81	0.86
8	0.89	0.81
9	0.90	0.79
10	0.83	0.85
11	0.78	0.86
12	0.69	0.86
13	0.52	0.87
14	0.49	0.87
15	0.88	0.83

Table 10-1 Initial Taylor and Bayliss Correlation Values

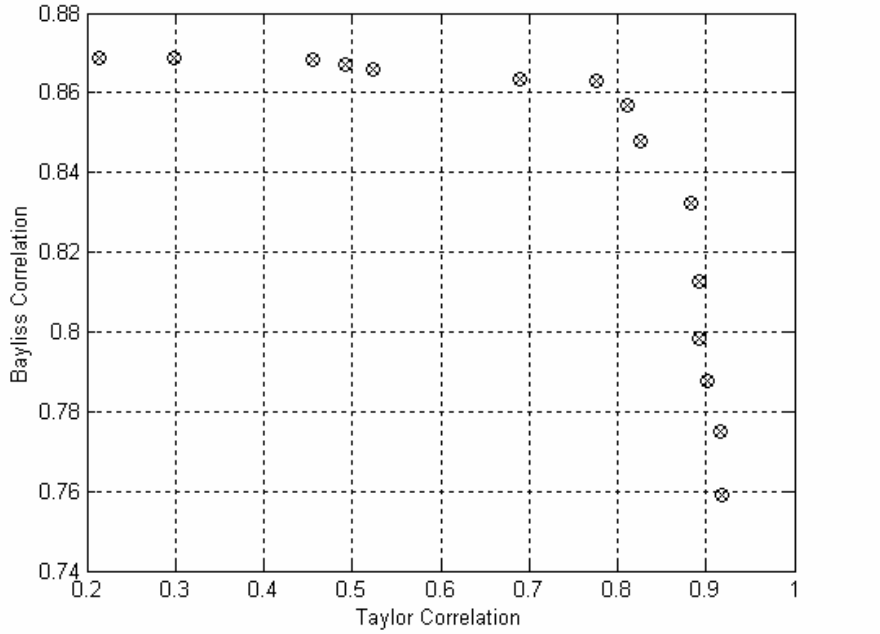


Figure 10-2 Correlation Coefficients of Optimised Tapers

It was necessary at this stage to evaluate the radiation pattern performance of the sum and multiple beam amplitude tapers. Four objective values were used to analyse the patterns, two for the sum pattern and two for the multiple beam pattern:

- f_1 = Mainlobe gain of the sum pattern;
- f_2 = Maximum sidelobe level in the sum pattern;
- f_3 = Average mainlobe gain in the multiple beam pattern.
- f_4 = Maximum sidelobe level in the multiple beam pattern;

The results set was then sorted to remove dominated solutions; the final set is shown in Table 10-2.

Solution	Taylor Correlation Coefficient	Bayliss Correlation Coefficient	f_1	f_2	f_3	f_4
2	0.919	0.759	55.298	23.720	50.961	14.565
3	0.893	0.798	53.301	24.233	51.443	12.623
5	0.214	0.869	57.969	10.104	51.029	20.391
6	0.455	0.868	58.092	15.480	51.354	17.869
8	0.891	0.813	54.312	24.647	51.178	15.395
9	0.901	0.788	53.235	24.685	51.179	12.855
10	0.826	0.848	56.083	15.594	51.588	18.095
11	0.776	0.863	56.340	13.819	51.169	18.718
12	0.689	0.863	56.655	13.034	51.106	18.688
13	0.523	0.866	57.921	15.764	51.118	17.910
14	0.493	0.867	57.957	15.591	51.223	18.459
15	0.882	0.832	54.279	24.381	50.414	17.798

Table 10-2 Radiation Pattern Performance

Analysis of these results show a close link between high correlation values and good pattern performance in both the sum and multiple beam cases.

Much of this thesis has been about describing and providing a choice of final optimised solutions - the remaining question is that now the Pareto set has been described, how do we use it?

I suspect the answer will be dependent on the intended use of the antenna. If the system requires frequent use of sum and multiple beam modes, then a solution that offers a good compromise across objectives would be a good choice. Solutions eight or ten are good 'all round performers' that offer a good compromise. Performance in any single objective would not be exceptional, but would be good and acceptable. If however the multiple beam pattern is only used occasionally, sum performance may be optimised at the expense of multiple beam performance (suggests solution nine would be acceptable).

Of the remaining solutions, solution five offered the lowest sidelobe multiple beam pattern performance, but poor sum performance. The optimised sum and multiple beam tapers and full array partitioning for solution five are shown below in Figure 10-3 and Figure 10-4.

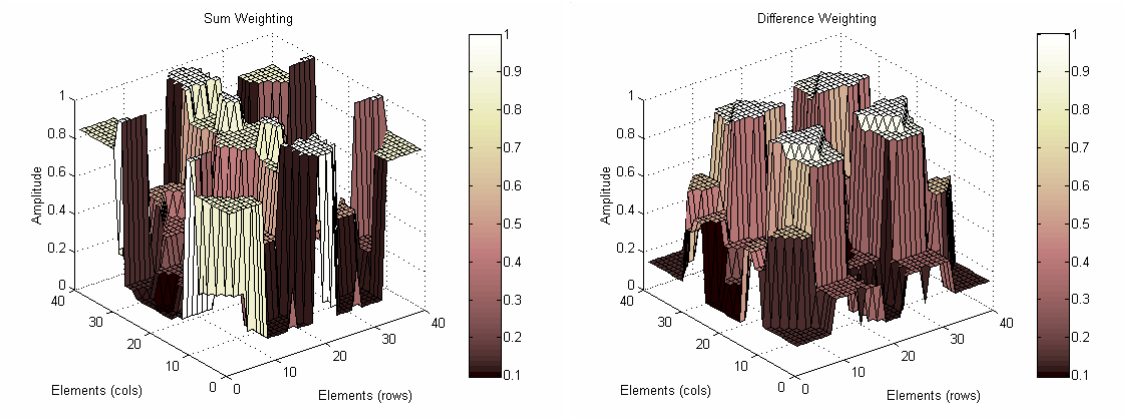


Figure 10-3 Optimised Sum and Difference Tapers

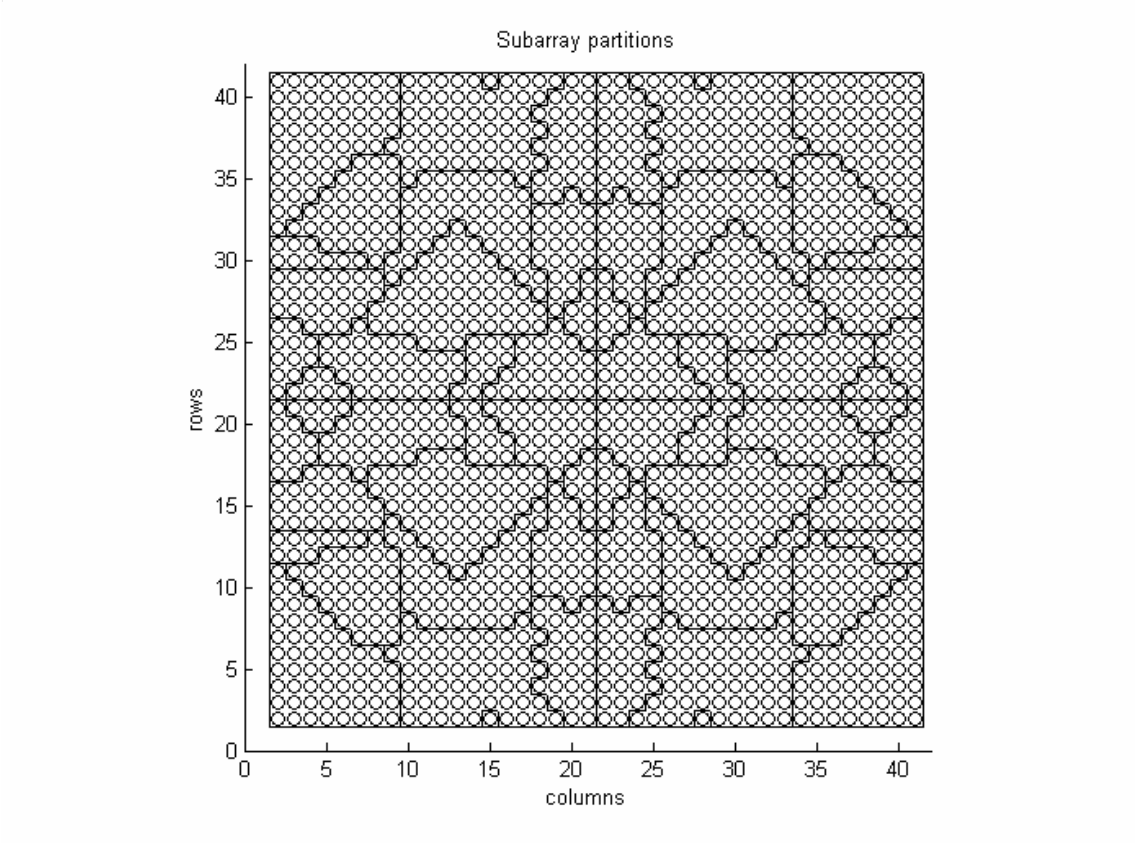


Figure 10-4 Optimal Partitioning for Sum and Difference Compromise

While the sum taper offered poor correlation with the Taylor, it was thought that changing a small number of the amplitude weights without altering the partitioning, could improve the correlation and hence the radiation pattern performance.

A simple local search algorithm was written to select one of the sixteen sum amplitude weightings and to increase or decrease it by 0.05. Any improvements in the correlation were saved, and degradations ignored (effectively a hill-climbing algorithm). After a further 100000 iterations, the correlation coefficient was increased from 0.214 to 0.937. The computation time was extremely fast as no antenna radiation pattern calculations were required to guide the optimisation process. The partitioning was not altered in any way. Figure 10-5 shows the improved sum taper (post local search).

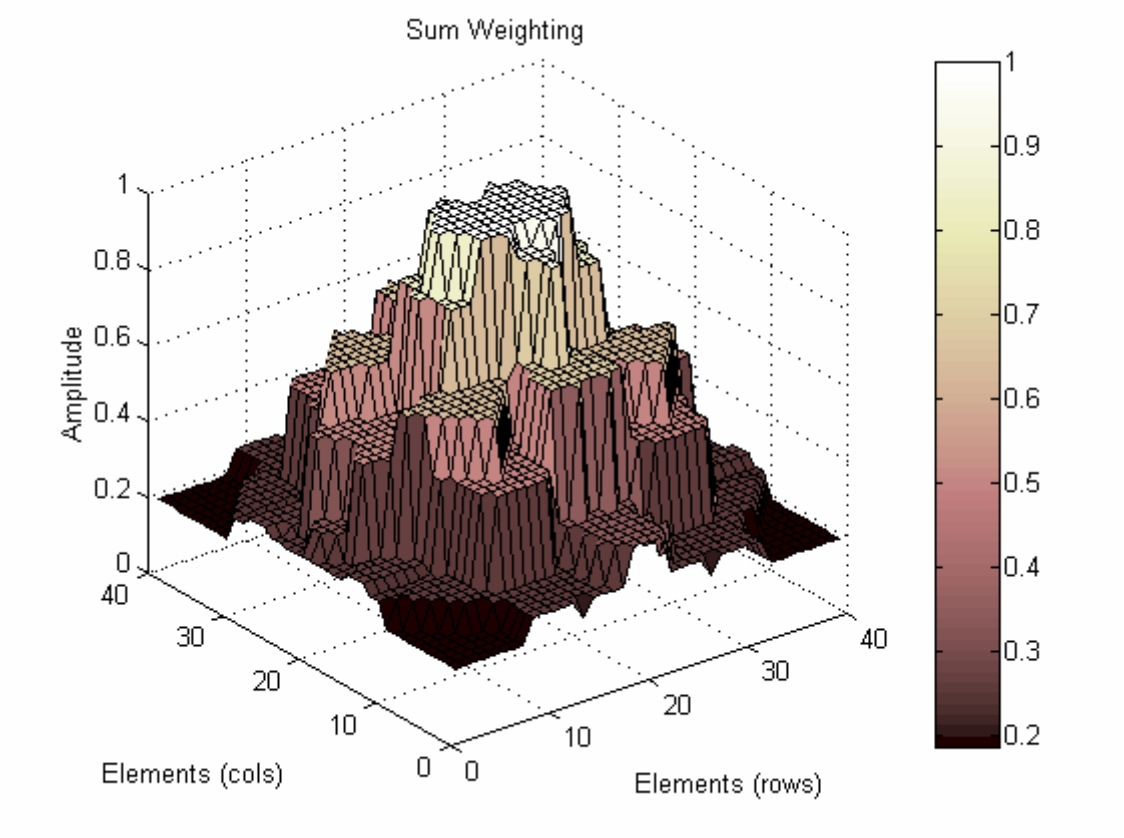


Figure 10-5 Sum Taper (Post Local Search)

The improved sum taper improved the sum radiation pattern performance to achieve mainlobe gain of 56.93dB and a maximum sidelobe level of -26.11dB. Figure 10-6 to Figure 10-9 show the radiation pattern performance for both the sum and difference cases.

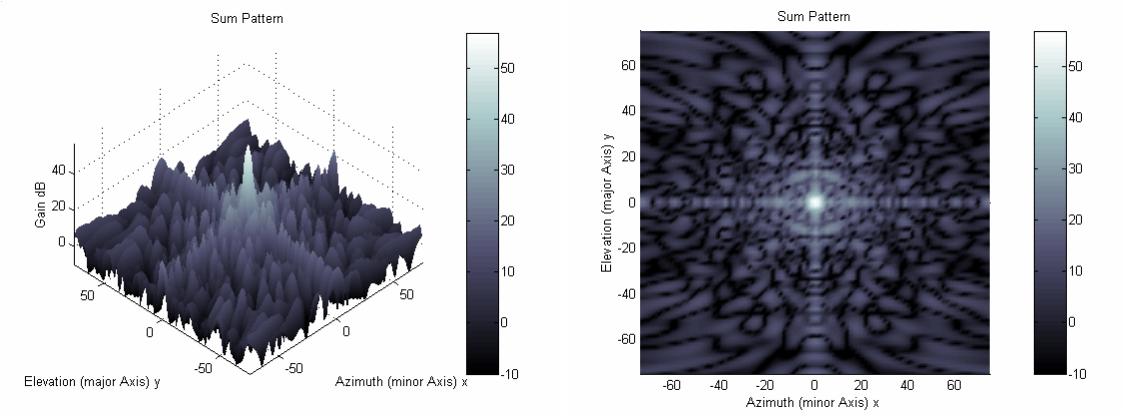


Figure 10-6 Radiation Pattern Performance (Sum Pattern)

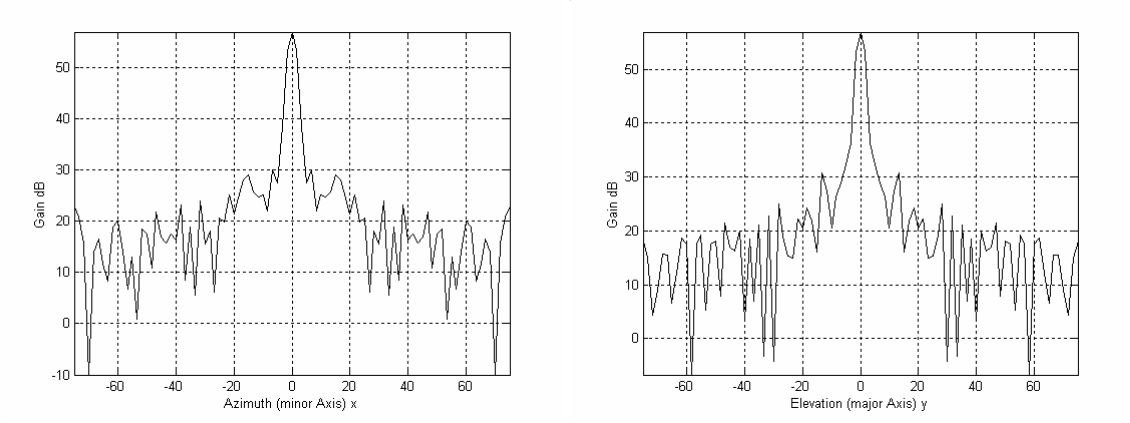


Figure 10-7 Sum Pattern Azimuth and Elevation Cuts

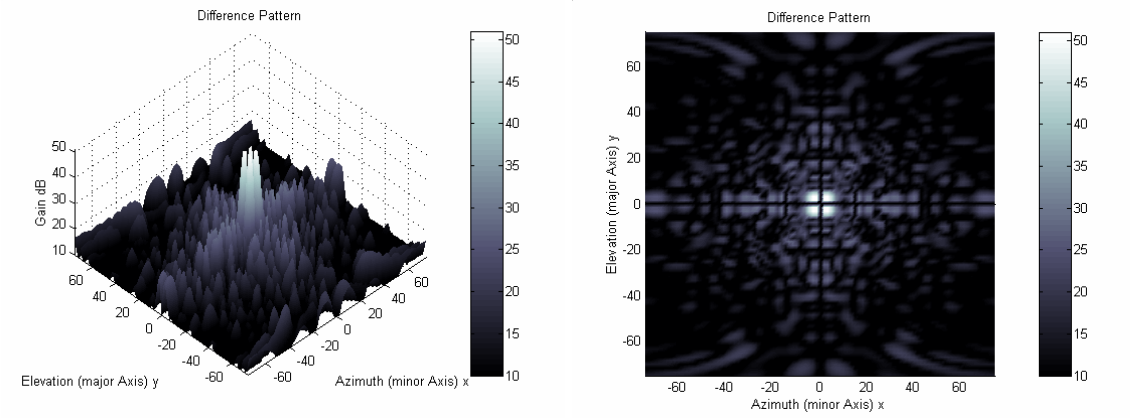


Figure 10-8 Radiation Pattern Performance (Difference Pattern)

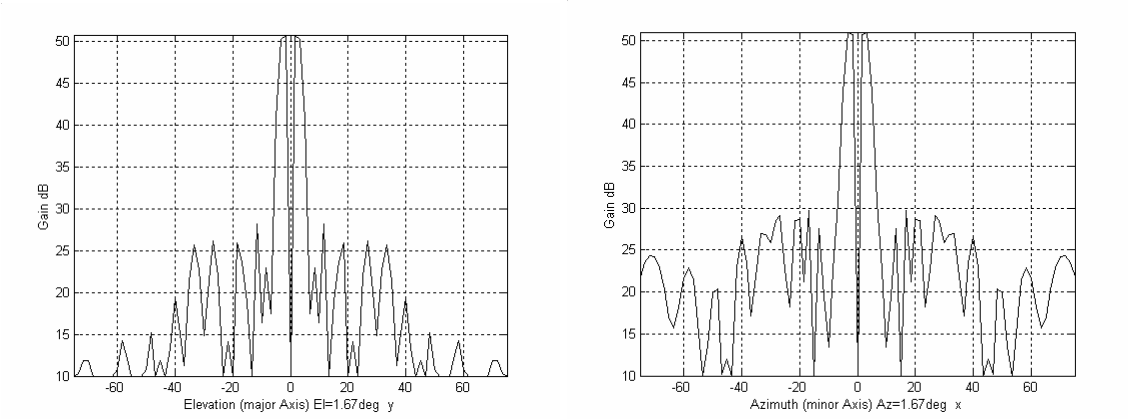


Figure 10-9 Difference Pattern Azimuth and Elevation Cuts

Table 10-2 was then updated to reflect the new improved solution five. The updated table is shown below:

Solution	Taylor Correlation Coefficient	Bayliss Correlation Coefficient	f_1	f_2	f_3	f_4
2	0.919	0.759	55.298	23.720	50.961	14.565
3	0.893	0.798	53.301	24.233	51.443	12.623
5 (old)	**0.214**	**0.869**	**57.969**	**10.104**	**51.029**	**20.391**
5 (new)	**0.933**	**0.896**	**56.93**	**26.110**	**51.029**	**20.391**
6	0.455	0.868	58.092	15.480	51.354	17.869
8	0.891	0.813	54.312	24.647	51.178	15.395
9	0.901	0.788	53.235	24.685	51.179	12.855
10	0.826	0.848	56.083	15.594	51.588	18.095
11	0.776	0.863	56.340	13.819	51.169	18.718
12	0.689	0.863	56.655	13.034	51.106	18.688
13	0.523	0.866	57.921	15.764	51.118	17.910
14	0.493	0.867	57.957	15.591	51.223	18.459
15	0.882	0.832	54.279	24.381	50.414	17.798

Table 10-3 Results Post Local Search

Post local search, solution five has the highest correlation performance, completely dominating all other solutions (when considering only correlation objective values). The radiation pattern objective measures show that it remains a non-dominated solution. In the absence of a secondary local search, solution 5 would not of been selected.

The use of a gradient search optimiser after the EA, effectively created a hybrid optimisation algorithm. The performance on this example was very good and so there was little advantage to be gained by experimenting with the methodology. The use of the secondary optimiser in this example worked because the objective functions are quick and easy to calculate. In problems with more computationally intensive functions such as the earlier examples in this thesis that require calculation of the entire radiation pattern, alternative approaches such as 'memetic algorthms' [155] are worthy of further research in multiobjective antenna problems. A memetic algorithm extends the search of a GA by applying a local search heuristic to improve performance. This search heuristic may take the form of a local gradient search applied to solution before the crossover and mutation genetic operators are applied.

10.3. Multiple Beams for Satellite Communications

10.3.1. Problem Statement

In this second case study, the performance of a low-earth-orbit, active phased-array satellite is optimised. The satellite array operates at 6 GHz and it used for communications functions.

The array was built with separate phase shifters for each element. The phase control provides a high degree of control to maximise the flexibility (and hence revenue) generated by the satellite during its operational life. Beam-shape flexibility greatly reduces the business-plan risk associated with a satellite, as it is possible to reconfigure the coverage area to respond to changing markets.

The satellite is to be leased to the military to provide secure communications between a number of geographically distance ground terminals. The military customer has specified that three independent beams can cover the ground terminals. A chart has been provided that details the required coverage relative to the array's boresight. The chart is shown in Figure 10-10, and it details three specific regions that must be illuminated by the antenna beams. The customer has also specified that the minimum acceptable EIRP in each region is 38dBW (EIRP).

The array contains 324 (18 x 18) elements arranged in a square grid. The manufacturers have stated that the element patterns can be approximated using a cos(theta) pattern.

To maximise the reliability of the array when space borne, amplitude control has been omitted and the TRMs operate at their nominal and most efficient power levels (uniform illumination). Control of the beam footprints and sidelobe levels must be carried out using phase control only.

The performance of the array in this application will be judged by the strength and uniformity of the three generated beams and by the relative sidelobe level.

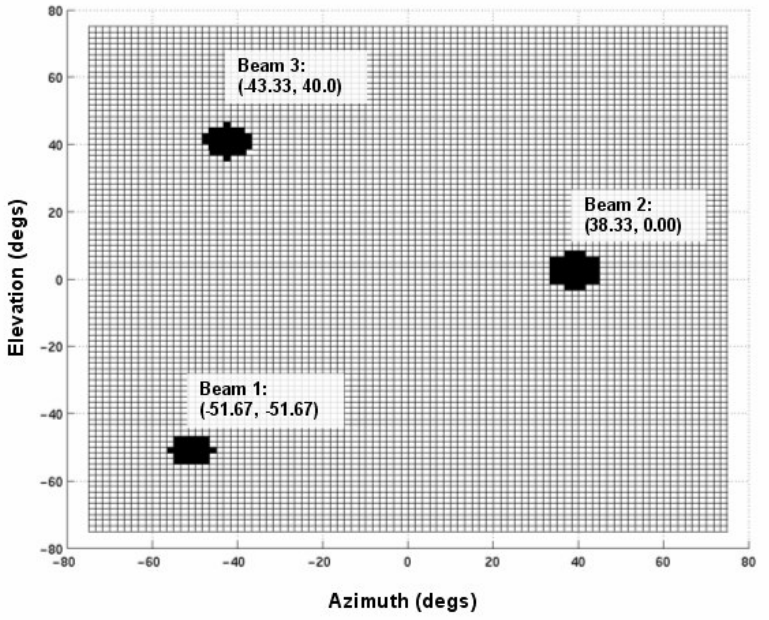


Figure 10-10 Required Multiple-Beam Coverage Regions

10.3.2. Solution

This array is quite small, and so generating three beams with low sidelobe levels will be challenging. If the elements were to be divided equally amongst the three beams, there would only be 108 elements contributing to any one beam so meeting the specified EIRP will be difficult.

In keeping with the hypothesis of this research, this problem will be solved using an EA and a method that is independent of the number of elements in the array:

The first step is to calculate three phase tapers that if applied individually, would form a single beam focused on one of the required regions. These tapers were calculated using Eq. 2-5 in Chapter 2. The problem now becomes one of determining an optimum method of assigning the phase values to the array's elements in such a way, that three uniform magnitude beams of high enough gain are formed, and the sidelobe level is acceptably low.

This problem was solved using the binary partitioning algorithm to divide the array into three 'virtual' subarrays. The elements in each of these three subarrays were in turn phased using sections of the original calculated phase values (see Figure 10-11). The net result is that each subarray generates a mainlobe steered to the desired ground location.

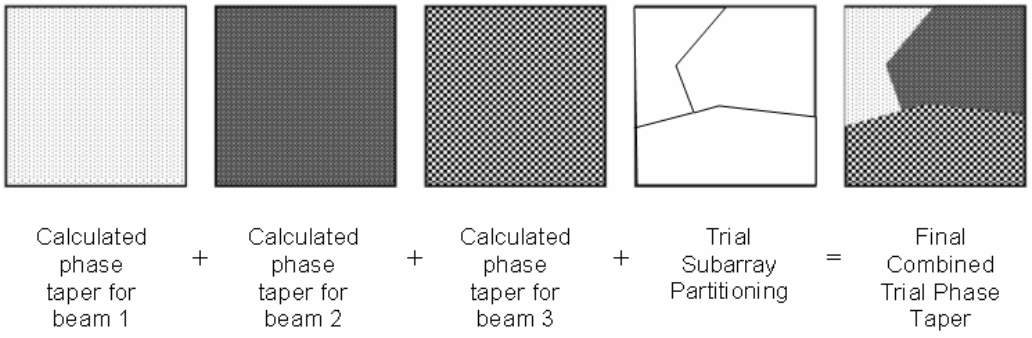


Figure 10-11 Generation of Trial Phase Taper

The shape and size of the three subarrays determines how many elements contribute to the formation of each beam, directly influencing the gain of each beam.

A number of objective measures were used to promote the desired characteristics in the radiation pattern. With reference to Figure 10-10 above, the radiation pattern was calculated using 1.66º resolution (91 x 91 samples). The three beam regions as defined above contained 35, 41 and 49 samples respectively. The gain of each sample in the three regions was analysed to determine the maximum in each region and these were designated G_1, G_2 and G_3. To promote the formation of high gain beams, the first objective measure f_1 was calculated using:

$$f_1 = G_1 + G_2 + G_3 \qquad \textbf{Eq. 10-1}$$

To encourage uniformity in the three beams the second objective measure f_2 was calculated using:

$$M = max(G_1 + G_2 + G_3) \qquad \textbf{Eq. 10-2}$$

$$f_2 = - [(M - G_1) + (M - G_2) + (M - G_3)] \qquad \textbf{Eq. 10-3}$$

A third objective measure f_3 measured the maximum sidelobe level relative to M:

$$f_3 = M - [\ max\ (sidelobe\ magnitudes\ )\ ] \qquad \textbf{Eq. 10-4}$$

A fourth and final objective measure f_4 minimised the maximum power in all regions except the desired beam regions encouraging the majority of the energy into the three mainlobe regions. The fourth objective also has the secondary effect of helping to reduce the beamwidth of each individual beam.

Cost functions formulated in this manner can produce strongly non-dominated solutions (the diversity in the solution set is less important in this application). There are advantages to reducing the number of objectives as it improves run times (particularly when performing non-dominated sorts), and helps to prevent the (EA specific) problem of all solutions becoming non-dominated and the algorithm being guided purely by measures of diversity in the solution set.
The ε-MOEA algorithm was used with an initial population size of 200 trial solutions and a further 10000 trial solutions were used to guide the evolutionary process. Using the binary fission partitioning algorithm ensured that only twelve variables needed optimising (hence the relatively small number of trial solutions). Analysis of the computational load showed that 95% of the time was spent evaluating the cost function (it took approximately one second to calculate each radiation pattern on a 2GHz PC).

The final non-dominated set contained 26 solutions. The solutions with the most uniform gain levels across the three mainlobes had poor sidelobe performance. Table 10-4 shows the non-dominated set in terms of the three gain levels and relative sidelobe levels. The results are sorted by sidelobe level in descending order (worst first):

Ref:	G_1:	G_2:	G_3:	SII
8	37.3767	38.8735	38.5706	-1.0329
9	39.1736	39.874	40.604	-6.8731
14	38.6534	40.8424	40.8944	-8.0939
6	38.8792	38.5455	38.5118	-8.71
15	40.8915	42.0123	38.9073	-8.9356
1	42.2935	40.8448	38.3811	-9.3921
23	42.1982	41.9595	37.2577	-10.4884
2	40.3295	40.2971	40.4041	-10.5099
4	42.1277	42.0058	37.2022	-10.517
26	40.1063	40.7091	40.2891	-10.9526
7	38.3328	40.9246	40.7833	-11.0134
19	38.0197	41.2266	40.6901	-11.1919
12	38.101	42.7777	40.4246	-11.8074
5	38.023	41.3955	41.3391	-12.0736
25	40.0171	40.9737	40.1435	-12.4243
16	41.1734	40.0823	39.941	-12.6228
22	39.7045	41.194	40.0982	-12.7585
20	40.8882	43.3502	36.3692	-12.7991
18	39.549	41.6425	39.2145	-13.0359
11	39.9244	41.8088	38.9609	-13.6632
3	39.8387	41.8829	38.8878	-14.1823
17	41.8676	42.8046	35.7089	-14.2674
13	38.8923	44.0389	37.015	-14.5392
21	38.2603	43.2117	38.4978	-14.7577
10	30.3106	47.4545	35.3195	-14.9279
24	31.3713	48.2089	32.3148	-15.3176

Table 10-4 Phase Taper Optimisation Results

Clearly a number of the solutions do not meet the specified EIRP levels and these can be discounted. However, the results suggest that if some minor differences in the magnitude of the three beams can be tolerated, the sidelobe level can be reduced by up to 4.8dB compared with one of the relatively uniform solutions (solution ref 2).

By way of example, solution ref. 3 is illustrated in the plots below, as it meets the specified ground terminal power level requirements (38 dBW minimum in each area) and has good sidelobe performance.

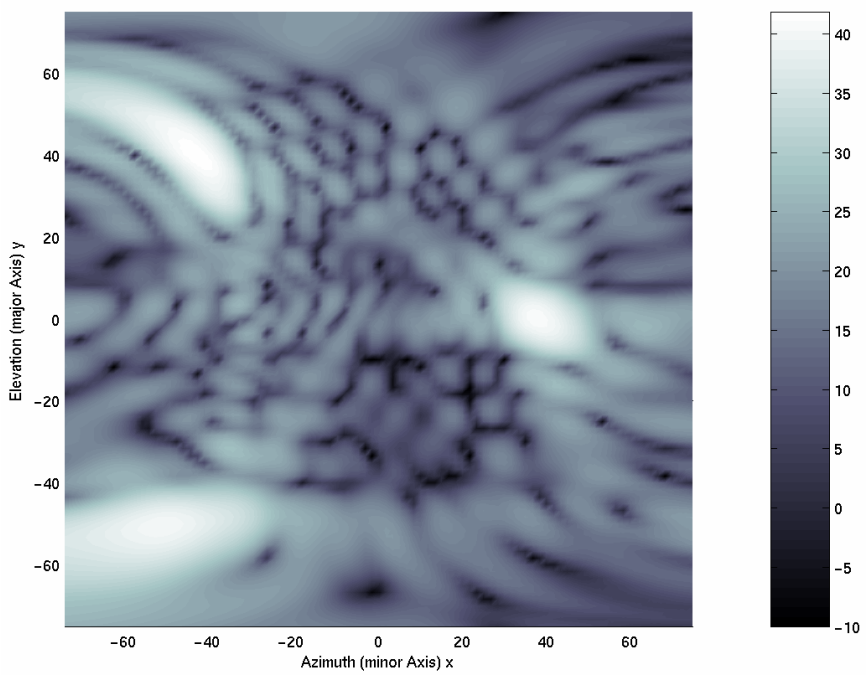


Figure 10-12 Radiation Pattern for Solution 3

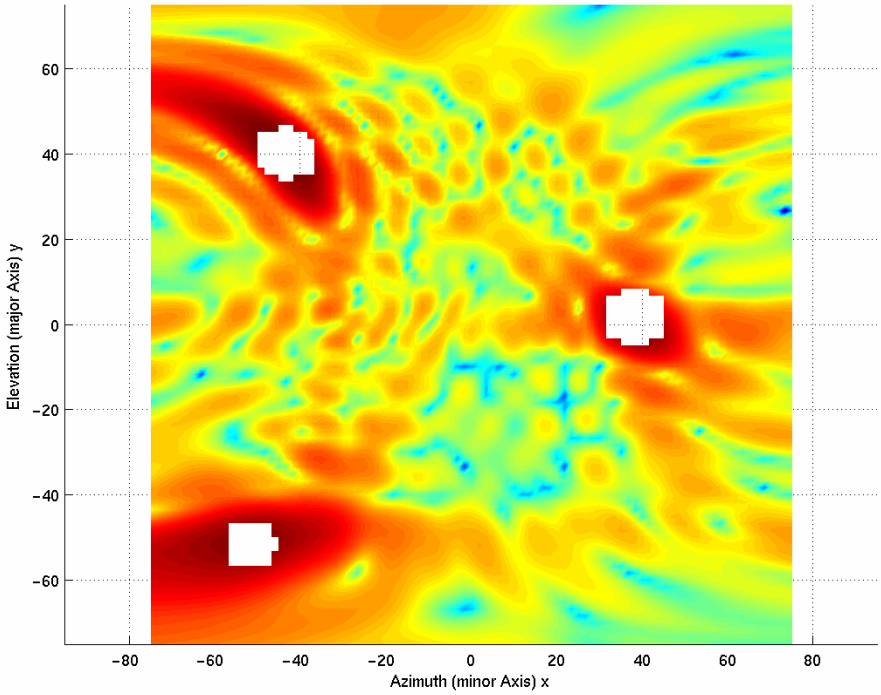


Figure 10-13 Compliance With Specified Coverage Areas

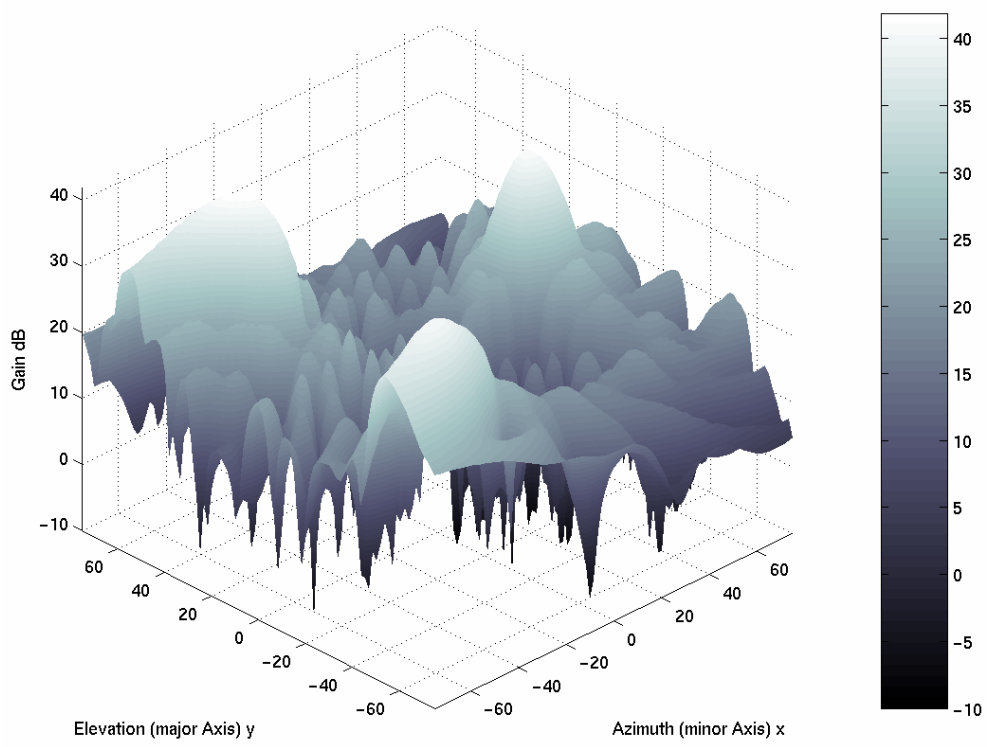


Figure 10-14 Three-Dimensional Radiation Pattern (Solution 3)

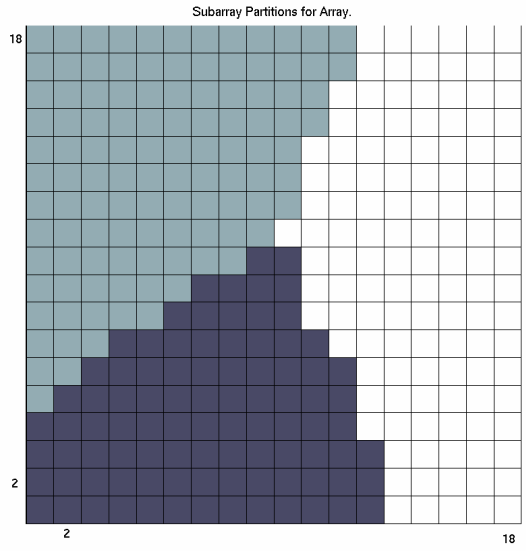


Figure 10-15 Optimised Partitioning (Solution 3)

CHAPTER 11
Conclusions

11. Conclusions

The amount of literature reviewed throughout the course of this research demonstrates just how popular research into active phased array antennas has become of late. Large phased arrays are becoming the antennas of choice in many applications but the cost of these arrays means that end users demand high levels of performance to justify the price premium. In order to achieve the performance, antenna designers are turning to new and novel methods for performance optimisation.

Similarly the interest in modern multiobjective evolutionary optimisation algorithms is another prevalent area of research. MOEAs have become very popular, very quickly and their predecessors such as the simple genetic algorithm, remain very popular optimisation tools of the electromagnetic research community. At the time of writing, modern evolutionary algorithms have been applied to array optimisation for just over ten years, but there has been little progress in this area in the last few years with no recent applications of multiple objective optimisation techniques in this area.

This thesis set out to correct this shortfall by exploring and furthering the application of modern evolutionary algorithms in the area of antenna array control. The research has provided new insight into the true nature of the solution spaces evident in array control problems and showed for the first time, the results of an exhaustive search of radiation patterns available from a small array and the set of true Pareto-optimal solutions.

It has been shown how the application of EAs in array control problems can be improved through new genotypic encoding methods such as the new method of encoding amplitude tapers using sine wave superposition. Problems with existing methods of analysis of radiation patterns have been discussed and improvements suggested. Importantly, the methods for genotypic encoding are independent of the number of elements in the antenna array. The encoding methods suggested are shown to work on large arrays (1600 elements) and prevent the array size limiting the performance and application of evolutionary algorithms.

A critical review of modern MOEAs has been completed which has highlighted the convergence and diversity performance of a number of leading, highly cited algorithms when applied to representative array control problems.

A new and novel algorithm has been developed that is inspired by the biological process of binary fission. It has been demonstrated how this algorithm can be used to partition array antennas into subarrays, irrespective of their geometry and how the partitioning process and hence the radiation pattern can be optimised using modern evolutionary algorithm. This part of the work was subject to two Patent applications in December 2003.

It was also demonstrated how multiobjective evolutionary algorithms could improve the performance of conformal array antennas. The application of the new binary fission partitioning algorithm was applied a conformal array and produced good results. This work was presented at the IEE's 12th International Conference on Antennas and Propagation in April 2003.

Finally a number of case studies show the diversity of the application of multiobjective evolutionary algorithms to solve examples of optimal compromise between sum and difference patterns, and to generate multiple beam radiation patterns for satellite applications.

Correct application of MOEAs were found to produce highly non-dominated sets of solutions in a computation time comparable with earlier generation EAs that only produced a single solution in the same run time. The MOEA prevents the antenna designer having to choose weights or some other method of combining antenna performance objectives in a single cost function.

Evolutionary approaches are not expected to find the best possible solution but instead they are able to find very good solutions. In many array control and partitioning problems, methods of completely solving the problem are unknown or not applicable due to lack of computational power. In antenna array control a very good solution is feasible as a result, and perfectly acceptable. It is shown in this work that multiple objective EAs can produce very strongly non-dominated solutions across the solution space that give the decision maker much more information with which to choose a final operating point for the array.

The good use of many modern optimisation methods is on problems that are not well enough understood for an analytical solution to be available. One advantage of EAs is that there is always the "current best solution". With classical optimisation methods we might not have access to the current best solution and on difficult problems such as those explored in this thesis, they may fail to converge at all.

CHAPTER 12 Recommendations for Future Work

12. Recommendations for Future Work

One of the limitations common to the Pareto-based multiple-objective evolutionary algorithms studied in this work, are their degradation in performance as the number of objectives increase. The number of non-dominated solutions increases significantly as the numbers of objectives are raised and in this event, the algorithms tend to be driven by diversity measures rather than convergence measures.

To use higher numbers of objectives (greater than seven or eight) on the types of antenna problems studied, the population size needs to be made much larger to ensure that at least some of the solutions become dominated by others. Increasing the population size reduces the rate of convergence and is especially bad practice in algorithms such as ε-MOEA that rely on frequent calculations to identify non-dominated solutions. Any further work spent evaluating the performance of new evolutionary algorithms would be beneficial, particularly if they were formulated to avoid the condition that occurs in the case of many non-dominated solutions existing.

Similarly, while MOEAs have been shown to produce good diversity in the solution set, in later generations after the convergence has slowed it has further been shown that strongly non-dominated solutions can be found through the application of a secondary local search algorithm. It may be advantageous and more computationally efficient to be able to task the original EA to focus on specific regions of interest in order to find more strongly non-dominated solutions from a smaller region of search space.

The partitioning methods developed in this work have been developed completely independently of array size. By weighing the subarrays uniformly, a large proportion of the solution space goes unexplored. More research effort would be welcomed that explores element level control of very large arrays in order to provide strongly non-dominated solutions closer to the true Pareto optimal operating points.

The application of EAs in real time applications such as for the correction of element failures in airborne arrays is some way off due to the processing time required to compute new excitation sets. Further work on fast converging evolutionary algorithms (such as micro-EAs) that would offer an improved level of performance, but not necessarily an optimum solution would be of benefit in the event of element failure.

Chapter 10 showed an example of the improvement of an EA results set through use of secondary local search. The use of memetic algorithms in computationally intensive multiobjective problems (such as large antenna array optimisation) is worthy of more research - it may improve convergence and reduce run times.

References

References

1. R.C. Hansen, "Phased Array Antennas", *Textbook, Wiley-Interscience, ISBN: 047153076X (1998).*
2. B.A. Kopp, M. Borkowski, G. Jerinic, “Transmit/Receive Modules”. *IEEE Trans. On Microwave Theory and Techniques. Vol 50, No 3 (March 2002).*
3. R.C. Hansen, ed., "Conformal Antenna Array Design Handbook". *Dept. of the Navy, Air Systems Command, (September, 1981).*
4. T.T. Taylor, “Design of Line-source Antennas for Narrow Beamwidth and Low Sidelobes”, *IRE Trans., vol AP-3, pp 16-28, (January, 1955).*
5. C.L. Dolph, “A Current Distribution for Broadside Arrays Which Optimises the Relationship between Beamwidth and Sidelobe Level”, *Proc. IRE, vol 34, pp 335-348, June, (1946).*
6. C.J. Drane, “Dolph-Chebyshev Excitation Coefficient Approximation”*, IEEE Trans., vol.AP-12, pp. 781-782, November, (1964).*
7. S.A, Schelkunov, "A Mathematical Theory of Linear Arrays", *Bell System Tech. J., pp 80-107, (1943).*
8. P.M. Woodward, "A Method of Calculating the Field Over a Plane Aperture Required to Produce a Given Polar Diagram", *Proc IEE, Part IIIA,Vol. 93, pp. 1554-1555 (1947).*
9. I.J.Gupta, A.A. Ksienski, “Effect of mutual coupling on the performance of adaptive arrays” *IEEE Trans. on Antennas and Propagation , Vol 31, pp 785-791, (Sept, 1983).*
10. C.A Band, C.D. Finlay, A.M. Kinghorn, R.W. Lyon, M.J. Smith, “Impact of

radomes on airborne pulse-Doppler radar performance" *Radar 2002 (IEE Conf. Publ. No.490), p 291-4 (2002).*
11. N. Metropolis, S. Ulam, "The Monte Carlo Method", *Journal of the American Statistical Association 44, N 247, 335-341, (1949).*
12. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, "Equation of State Calculations by Fast Computing Machines," *J. Chem. Phys. 21 1087-1092, (1953).*
13. S. Kirkpatrick, C.D. Gellatt Jr., M.P. Vecchi, "Optimisation by Simulated Annealing" *Volume 220, No 4598, SCIENCE Journal, (13 May 1983).*
14. A Bjork "Numerical Methods for Least Squares Problems", *SIAM, Philadelphia, ISBN 0-89871-360-9, (1996).*
15. O.M Bucci, G. D'Elia, G. Romito "A Generalised Projection Technique for the Synthesis of Conformal Arrays" *Antennas and Propagation Society International Symposium, 1995. AP-S. Digest, Volume: 4, Page(s): 1986 -1989 vol.4 (1995).*
16. G.T Poulton, G.T, Hay, "Efficient design of shaped reflectors using successive projections." *Electronics Letters , Volume: 27 Issue: 23 , Page(s): 2156 -2158, (7 Nov. 1991)*
17. G.T. Poulton, "Power Pattern Synthesis Using the Method of Successive Projections", *1986 International Symposium Digest Antennas and Propagation (Cat. No.86CH2325-9), p 667-70 vol.2 (1986).*
18. J.A. Rodriguez, L. Landesa, J.L. Rodriguez, F. Obelleiro, F. Ares, A. Garcia-Pino, "Pattern synthesis of array antennas with arbitrary elements by simulated annealing and adaptive array theory". *Microwave and Optical Technology Letters, v 20, n 1, p 48-50. (5 Jan. 1999)*
19. P. López, J. A. Rodríguez, F. Ares, and E. Moreno, "Optimization of Aperture Distributions for Double-Difference Patterns", *IEEE Transactions On Antennas And Propagation, Vol. 51, No. 3, (March 2003).*
20. L. I. Vaskelainen, "Iterative least-squares synthesis methods for conformal array antennas with optimized polarization and frequency properties" *Antennas and Propagation, IEEE Transactions on , Volume: 45 Issue: 7 , Page(s): 1179 -1185 (July 1997).*
21 R.F.E. Guy, R.A. Lewis, P.J. Tittensor, "Conformal Phased Arrays", *First European Conference on Conformal Arrays, Karlsruhe 1999.*
22. L. Zadeh, "Fuzzy Sets" *Information and Control, 8:338-353, (1965).*
23. K. Min-Soeng, "Structure/parameter optimization of fuzzy models by evolutionary algorithm" *Proceedings of the IASTED International Conference on Intelligent Systems and Control, p 295-300 ISBN: 0 88986 355 5 (2003).*
24. J.T. Alander. "Indexed bibliography of genetic algorithms with fuzzy systems." *Report 94-1-FUZZY, University of Vaasa, Department of Information Technology and Production Economics, (1995).*
25. R. Beale, T Jackson, "Neural computing an introduction" *ISBN 0-85274-262-2 (1990).*
26. R.P. Lippman "An intoduction to computing with neural nets", *IEEE ASSP Magazine (April 1987).*
27. T. Kohonen, "An Introduction to Neural Computing. In Neural Networks", *volume 1, number 1, (1988).*
28. M.A. Aboul-Dahab, K.A Hijjah, S.E. El-Khamy, "A new technique for linear antenna array processing for reduced sidelobes using neural networks.",

Proceedings of the Fifteenth National Radio Science Conference. NRSC '98 (Cat. No.98EX109), p B1/1-8, ISBN: 0 7803 5121 5 (1998).

29. S. Reza, C.G. Chrostodulou, "Beam shaping with antenna arrays using neural networks", *Southeastcon '98. Proceedings. IEEE p 220-223 (24-26 Apr 1998).*
30. D. Goldberg, "Genetic Algorithms", *Addison-Wesley, (1989).*
31. J. Holland, "Adaption in Natural and Artificial Systems", *Ann Arbor, The University of Michigan Press, (1975).*
32. L.J. Fogel, A.J Owens, M.J. Walsh, "Artificial Intelligence through Simulated Evolution", *New York: Wiley (1966).*
33. D. B. Fogel (Editor) "Evolutionary Computation : The Fossil Record", Text book, John Wiley & Sons; (May 1, 1998) ISBN: 0780334817
34. R. Storn, K Price, "Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces", *Technical Report, Number: TR-95-012, University of Berkley, Berkeley, CA, USA, (1995).*
35. J.R. Koza, "Hierarchical Genetic Algorithms Operating on Populations of Computer Programs." *Proc.11th International Joint Conference on Artificial Intelligence (IJCAI). San Mateo: Morgan Kaufman (1989).*
36. J.R. Koza, "Genetic Programming: A Paradigm for Genetically Breeding Populations of Computer Programs to Solve Problems". *Report, Stanford University Computer Science Departments Technical Report STAN-CS-90-1314. (June 1990).*
37. I. Rechenberg, "Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution", *Frommann-Holzboog, Stuttgart, (1973).*
38. H-P. Schwefel, "Evolutionsstrategie und Numerische Optimierung", *Dissertation, Technical University of Berlin and Medical University of Hannover, (1975).*
39. H-P. Schwefel, "Binäre Optimierung durch Somatische Mutation", *Technical Report, Technical University of Berlin and Medical University of Hannover, (1975).*
40. H-P. Schwefel, "Collective Phenomena in Evolutionary Systems", *Preprints of the 31st Annual Meeting of the International Society for General System Research, Budapest 2, 1025-1033, (1987).*
41. S. Baluja, "Population-Based Incremental Learning: A Method for Integrating Genetic Search Based Function Optimization and Competitive Learning", *Technical report, CMU-CS-94-163", School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, (1994).*
42. E.J. Hughes, "Optimisation Using Population Based Incremental Learning (PBIL)", *Optimisation in Control: Methods and Applications (Ref. No. 1998/521), IEE Colloquium on, Pages:2/1 - 2/3, (10 Nov. 1998).*
43 J. Kennedy, R. Eberhart, "Particle Swarm Optimization", *1995 IEEE International Conference on Neural Networks Proceedings (Cat. No.95CH35828), Perth, WA, Australia, 1942-8 vol.4, IEEE 1995.*
44. Y. Fukuyama, H. Yoshida, "A Particle Swarm Optimization for Reactive Power and Voltage Control in Electric Power Systems", *Evolutionary Computation, 2001. Proceedings of the 2001 Congress on , Volume: 1 , 27-30 Pages:87 - 93 vol. 1 (May 2001)*
45. B. Al-kazemi, C.K Mohan, "Training Feedforward Neural Networks Using Multi-phase Particle Swarm Optimization", *Neural Information Processing, 2002. ICONIP '02. Proceedings of the 9th International Conference on, Volume: 5,*

Pages:2615 – 2619, (Nov. 18-22, 2002).

46. A.A.A. Esmin, A.R. Aoki, G. Lambert-Torres, "Particle swarm optimization for fuzzy membership functions optimization", *Systems, Man and Cybernetics, 2002 IEEE International Conference on, Volume: 3, Pages: 6 pp. vol.3, (6-9 Oct. 2002*).
47. M. Dorigo, V. Maniezzo, A. Colorni. "The Ant System: An Autocatalytic Optimizing Process.", *Technical Report No. 91-016 Revised, Politecnico di Milano, Italy (1991).*
48 W.H. Press, B.P.Flannery, S.A.Teukolsky, W.T.Vetterling, "Numerical Recipes in C : The Art of Scientific Computing", *Cambridge University Press; 2 edition ISBN: 0521431085 Page 445 (October 30, 1992)*
49. M. Dorigo "Ant Colony Optimization" *Website url:* http://iridia.ulb.ac.be/~mdorigo/ACO/index.html
50. A.J. Champandard, "Artificial Intelligence AI Depot" *Website url:* http://ai-depot.com/CollectiveIntelligence/Ant-Colony.html
51. D.B. Fogel, "The Advantages of Evolutionary Computation", *Bio-Computing and Emergent Computation 1997, World Scientific Press, Singapore, pages 1-11, (1997).*
52. A. Akdagli, K. Guney, D. Karaboga, "Pattern nulling of linear antenna arrays by controlling only the element positions with the use of improved touring ant colony optimization algorithm", *Journal of Electromagnetic Waves and Applications, vol 16, n 10, p 1423-1441, (2002).*
53. A. Hoorfar, Z. Jinhui, "A novel hybrid EP-GA method for efficient electromagnetics optimization", *Antennas and Propagation Society International Symposium, 2002. IEEE, Volume: 1, Pages:310 - 313 vol.1, (16-21 June 2002).*
54. D.G Kurup, M**.** Himdi, A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm", *Antennas and Propagation, IEEE Transactions on, Volume: 51, Issue: 9, Pages:2210 – 2217, (Sept 2003).*
55. J.M Horrell, L.J. du Toit, "Array pattern synthesis using PBIL", *AFRICON, 1996., IEEE AFRICON 4th , Volume: 1 , Pages:276 - 281 vol.1, (24-27 Sept. 1996).*
56. D.W. Boeringer, D.H. Werner, "A comparison of particle swarm optimization and genetic algorithms for a phased array synthesis problem", *Antennas and Propagation Society International Symposium, 2003. IEEE , Volume: 1 , June 22-27, Pages:181 – 184, (2003).*
57. D.Gies, Y.Rahmat-Samii, "Reconfigurable array design using parallel particle swarm optimization", *Antennas and Propagation Society International Symposium, 2003. IEEE , Volume: 1 , Pages:177 – 180, (June 22-27, 2003).*
58. R. L. Haupt, J. J. Menozzi, C. J. McCormack. "Thinned arrays using genetic algorithms", *IEEE Antennas and Propagation Society, International Symposium 1993, volume 2, pages 712—715, (1993).*
59. D.J, O'Neill. "Element placement in thinned arrays using genetic algorithms" *OCEANS 94. Oceans Engineering for Today's Technology and Tomorrow's Preservation. Proceedings (Cat. No.94CH3472-8), 1994, pt. 2, p II/301-6 vol.2. (1994).*
60. M Shimizu. "Determining the Excitation Coefficients of an Array using Genetic Algorithms". *Antennas and Propagation Society International Symposium AP-S Digest, Volume 1, Pages 530-533. (51994)*
61. J.M Johnson, Y. Rahmat-Samii, "Genetic Algorithms in Engineering

Electromagnetics". *IEEE Antennas and Propagation Magazine, Vol.39, No 4, (August 1997).*

62. R. L. Haupt. "Comparison between genetic and gradient-based optimization algorithms for solving electromagnetics problems" *IEEE Transactions on Magnetics, v 31, n 3, p 1932-5. (May 1995).*
63. R. L. Haupt. "Optimization of Subarray Amplitude Tapers". *Antennas and Propagation Society International Symposium, AP-S. Digest, Volume:4, pages 1830-1833 vol.4. (1995).*
64. D. Marcano, F, Duran, O.Chang, "Synthesis of Multiple Beam Linear Antenna Arrays Using Genetic Algorithms", *IEEE Antennas and Propagation Society International Symposium. 1995 Digest (Cat. No.95CH35814), pt. 2, p 938-41 vol.2 (1995).*
65. D. Marcano, M. Jimenez, F, Duran, O. Chang, "Synthesis of antenna arrays using genetic algorithms", *Proceedings of the 1995 First IEEE International Caracas Conference on Devices, Circuits and Systems (Cat. No.95TH8074), p 328-32 (1995).*
66. A. Alphones, V. Passoupathi, "Null steering in phased arrays by positional perturbations: a genetic algorithm approach", *1996 IEEE International Symposium on Phased Array Systems and Technology. Revolutionary Developments in Phased Arrays (Cat. No.96TH8175), p 203-7, (1996).*
67. F. Ares, S.R. Rengarajan, J.A. Ferreira1, A. Trastoy, **"**Synthesis of Antenna Patterns of Circular Arc Arrays" *IEE AP-S International Symposium and URSI North American Radio Science Meeting, (1997).*
68. D.S. Weile, E. Micheilssen, D.E. Goldberg, "Multiobjective synthesis of electromagnetic devices using nondominated sorting genetic algorithms", *IEEE Antennas and Propagation Society International Symposium. 1996, pt. 1, p 592-5 vol.1*
69. N. Srinivas, K. Debb, "Mulitobjective Optimisation Using Nondominated Sorting in Genetic Algorithms." *Evolutionary Computation, 2(3):221-243, (1994).*
70. D.S. Weile, E. Micheilssen, "Integer Coded Pareto Genetic Algorithm Design of Constrained Antenna Arrays", *IEEE, Electronics Letters, Vol 32. No. 19. (12 September 1996).*
71. C.W. Brann, K.L, Virga, "Generation of Optimal Distribution Sets for Single Ring Cylindrical Arc Arrays." *IEEE AP-S URSI International Symposium, Atlanta, Georgia, (June 22-26, 1998).*
72. M.V Lozano, J.A Rodriguez, F. Ares, "Recalculating linear array antennas to compensate for failed elements while maintaining fixed nulls" *Journal of Electromagnetic Waves and Applications, v 13, n 3, p 397-412 (1999).*
73. P. Lopez, J.A. Rodriguez, F. Ares, E. Moreno, "Low-sidelobe patterns from linear and planar arrays with uniform excitations except for the phases of a small number of elements", *IEEE Electronics Letters, Vol. 37, No.25. (6th December 2001).*
74. C.D Bregon, E. Fernandez del Rio, "Hybrid optimizer based on genetic algorithms and conjugate gradient", *IEEE Antennas and Propagation Society International Symposium (IEEE Cat. No.02CH37313), pt. 1, p 738-41 vol.1, (2002).*
75. M. Mitchell, R. Howard, C. Tarran, "Adaptive digital beamforming (ADBF) architecture for wideband phased array radars", *Tactical Sensors including Infrared and Radar, Proceedings of SPIE (1999).*

76. P. Lopez, J.A. Rodriguez, "Subarray Weighting for the Difference Patterns of Monopulse Antennas: Joint Optimisation of Subarray Configurations and Weights" *IEEE Transactions on Antennas and Propagation, Vol 49, No 11 (November 2001).*
77. D.W Ansell, E.J. Hughes, "Antenna Array", UK Patent Office Application Number 0229123.5, Filed 13th December 2003.
78. D.W Ansell, E.J. Hughes, "Partitioning Process for Antenna or Sensor Array", UK Patent Office Application Number 0229121.9, Filed 13th December 2003.
79. D.W Ansell, E.J. Hughes, "Use of multi-objective genetic algorithms to optimise the excitation and subarray division of multifunction radar antennas" *IEE Multifunction Radar and Sonar Sensor Management Techniques (Ref. No.01/173), p 8/1-4. (November 2001).*
80. D.W Ansell, E.J. Hughes, "Using multi-objective genetic algorithms to optimise the subarray partitions of conformal array antennas" *Twelfth International Conference on Antennas and Propagation. ICAP 2003 (IEE Conf. Publ. No.491), pt. 1, p 151-5 vol.1, (April 2003).*
81. R.F Harrington, "Field computation by moment methods", *Collier-Macmillan, New York, 229pp, 1968.*
82. J. Moore, R. Pizer, eds. "Moment methods in electromagnetics: techniques and applications", *Research Studies Press, ISBN 0863800130, 398pp 1984.*
83. E.K. Miller, L. Medgyesi-Mitschang, E.H. Newman, "Computational electro-magnetics: frequency domain method of moments", *IEEE Press, New York, ISBN 0879422769, 508 pp 1992*
84. S. Silver. "Microwave antenna theory and design", *MIT Radiation Laboratory Series, McGraw Hill Book Co., New York, Vol 20, 1949*
85. Y. Rahmat-Samii. "A comparison between GO/aperture field and physical optics methods for offset reflectors", *IEEE Trans Antennas Propagat., Vol AP-32, 1984, pp301- 306.*
86. P.F. Goldsmith, "Quasi-optical techniques at millimeter and sub-millimeter wavelengths in Infrared and Millimeter Waves", Vol 6, *1982, Academic Press, New York.*
87. J.B. Keller "A geometrical theory of diffraction" *In Calculus of variations and its applications, edited by Graves L.M., McGraw Hill Book Co., New York, 1958, pp27-52*
88. R.G. Kouyoumjian, P.H.Pathak, "A uniform geometrical theory of diffraction for an edge in a perfectly conducting surface", *Proc IEEE, Vol 62, 1974, pp1448-1461*
89. M. Skolnik, "Radar Handbook" *2nd Edition, Mc-Graw Hill, 1990.*
90. "IEEE standard definitions of terms for antennas.", *IEEE Std 145-1993 , 21 June 1993*
91. G.P. Junker, S.S. Kuo, C.H. Chen, "Genetic algorithm optimization of antenna arrays with variable interelement spacings", *Antennas and Propagation Society International Symposium, 1998. IEEE , Volume: 1 , 21-26 June 1998 Pages:50 - 53 vol.1*
92. I. Salonen, A. Toropainen, P. Vainikainen, "Linear Pattern Correction in a Small Microstrip Antenna Array", *Antennas and Propagation, IEEE Transactions on , Volume: 52 , Issue: 2 , Feb. 2004 Pages:578 - 586*
93. D.H. Werner, M.A. Gingrich, P.L. Werner, "A self-similar fractal radiation

pattern synthesis technique for reconfigurable multiband arrays", Antennas and Propagation, IEEE Transactions on , Volume: 51 , Issue: 7 , July 2003, Pages:1486 - 1498

94. D.G. Kurup, M. Himdi, A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm", *Antennas and Propagation, IEEE Transactions on , Volume: 51 , Issue: 9 , Sep 2003 Pages:2210 - 2217*
95. A. Trastoy, Y. Rahmat-Samii, F.Ares, E. Moreno, "Two-pattern linear array antenna: synthesis and analysis of tolerance", *Microwaves, Antennas and Propagation, IEE Proceedings , Volume: 151 , Issue: 2 , April 2004, Pages:127 - 130*
96. M.I. Skolnik, "Introduction to Radar Systems", *McGraw-Hill 2nd Edition , page 281.*
97. C.W. Brann, K.L Virga, "Generation of Optimal Distribution Sets for Single Ring Cylindrical Arc Arrays.", *IEEE AP-S URSI International Symposium (Atlanta, Georgia, June 22-26, 1998)*
98 H. Nyquist, "Certain topics in telegraph transmission theory," *Trans. AIEE, vol. 47, pp. 617-644, Apr. 1928.*
99 C. E. Shannon, "Communication in the presence of noise," *Proc. Institute of Radio Engineers, vol. 37, no.1, pp. 10-21, Jan. 1949.*
100. L.D. Davis "Bit Climbing, representational bias, and test suite design". *In R.K. Belew and L.B.Booker (eds.), Proceedings of the Fourth International Conference on Genetic Algorithms, 18-23. San Mateo, CA: Morgan Kaufmann.*
101. M. Mitchell, J.H. Holland, "When will a genetic algorithm outperform a hill climbing?", *Proceedings of the Fifth International Conference on Genetic Algorithms. Forrest, S. (ed.), Morgan Kaufmann, 647 (1993).*
102. C. Darwin, "On the Origin of Species by Means of Natural Selection", *London: J. Murray. (1859)*
103 D. E. Goldberg, K. Sastry, T. Latoza, "On The Supply Of Building Blocks", *Proceedings of the Genetic and Evolutionary Computation Conference ({GECCO}-2001)*
104. K. Krishnakumar, "Micro-Genetic Algorithms for Stationary and Non-Stationary Function Optimisation". *SPIE's Intelligent Control and Adaptive Systems Conf. Paper # 1196-32 (1989)*
105. W. M. Spears and K. A. De Jong, "An Analysis of Multi-Point Crossover", In *Foundations of Genetic Algorithms*, *J. E. Rawlins (Ed.), pp. 301-315, 1991.*
106. A. Neubauer, "The Circular Schema Theorem for Genetic Algorithms and Two-Point Crossover", *GALESIA 97, Glasgow 2-4 September 1997, IEE pp 209-214, 1997.*
107. G. Syswerda, "Uniform crossover in genetic algorithms", *Proc. ICGA 3, pp. 2-9, 1989.*
108. W. M. Spears and K. A. De Jong, "On the Virtues of Parameterised Uniform Crossover", *Proc. ICGA 4*, pp.230-236, 1991.
109. A. H. Wright, "Genetic Algorithms for Real Parameter Optimisation", In *Foundations of Genetic Algorithms*, *J. E. Rawlins (Ed.), Morgan Kaufmann, pp. 205-218, 1991.*
110. D. M. Tate and A. E. Smith, "Expected Allele Convergence and the Role of Mutation in Genetic Algorithms", *Proc. ICGA 5, pp.31-37, 1993.*

111. G. Syswerda, "A study of Reproduction in Generational Steady-State Genetic Algorithms.", *In Foundations of Genetic Algorithms, Rawlings G.J.E. (ed.), Morgan Kaufmann, San Mateo, CA.1991.*
112. K. A. De Jong and J. Sarma, "Generation Gaps Revisited", *In Foundations of Genetic Algorithms 2, L. D. Whitley (Ed.), Morgan Kaufmann Publishers, 1993.*
113. J.A. Rodriguez, P. Lopez, J.C. Bregains, F. Ares, E. Moreno, "Almost uniformly excited arrays", *Antennas and Propagation Society International Symposium, 2002. IEEE ,Volume: 1 , Pages: 322 - 325 vol.1 (16-21 June 2002)*
114. K. Chellapilla, A. Hoorfar "Evolutionary programming: an efficient alternative to genetic algorithms for electromagnetic optimization problems", *Antennas and Propagation Society International Symposium, 1998. IEEE ,Volume: 1 ,Pages:42 - 45 vol.1 21-26 June 1998*
115. D. Feng Li, C. Lin Gong, "Design of hexagonal planar arrays using genetic algorithms for performance improvement", *ICMMT 2000. 2000 2nd International Conference on Microwave and Millimeter Wave Technology Proceedings (Cat. No.00EX364), p 455-60, 2000*
116. Vilfredo Pareto. "Cours D'Economie Politique", *volume I and II. F. Rouge, Lausanne, 1896.*
117. Donnelly, W. Dawber, "Modelling and Optimisation of Adaptive MFRs", *IEE Multifunction Radar and Sonar Sensor Management Techniques (Ref. No.01/173), (November 2001).*
118. J. D. Schaffer. "Multiple objective optimization with vector evaluated genetic algorithms. In Genetic Algorithms and their Applications", *Proceedings of the First International Conference on Genetic Algorithms, pages 93–100. Lawrence Erlbaum, (1985).*
119. E. Zitler, "Evolutionary Algorithms for Multiobjective Optimisation: Methods and Applications." *PhD thesis, Swiss Federal Institute of Technology Zurich (ETH), 1999. Diss ETH No. 13398.*
120. C.M. Fonseca, P.J. Fleming, "Genetic Algorithms for Multiobjective Optimisation: Formulation, Discussion and Generalisation." *Proceedings of the Fifth International Conference on Genetic Algorithms, pages 416-423, San Mateo, California, 1993. University of Illinois at Urbana-Champaign, Morgan Kauffman Publishers.*
121. K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, "A Fast Elitist Non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II", Proceedings of the Parallel Problem Solving from Nature VI Conference , 16-20 September. (Paris, France), (pp. 849--858). (2000)
122. K. Deb, T. Goel, "Controlled Elitist Non-dominated Sorting Genetic Algorithms for Better Convergence" KanGAL Report No. 200004 (2000) available online at url: http://www.iitk.ac.in/kangal/papers/tech-rep4.ps.gz
123. Summarised Publication and Citation Data from ISI for the Analysis of Research Trends & Performance url: http://www.esi-topics.com/
124. J. Horn, N. Nafpliotis, D.E. Goldberg, "A Niched Pareto Genetic Algorithm for Multiobjective Optimization." Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on , Page(s): 82 -87 vol.1 (1994)
125. D.E. Goldberg, J. Richardson,"Genetic Algorithms with Sharing for Multimodal Function Optimization". *In Grefenstette, editor, Genetic Algorithms and their*

Applications (ICGA'87), pages 41--49.(1987)
126. J.D. Knowles, D.W. Corne. "The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation." *In Congress on Evolutionary Computation (CEC99), volume 1, pages 98. IEEE Press, (1999).*
127. E. Zitzler, L Thiele "An Evolutionary Algorithm for Multiobjective Optimization: The Strength Pareto Approach". Technical Report 43, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 1998. url: http://citeseer.ist.psu.edu/article/zitzler98evolutionary.html
128. E. Zitzler, M. Laumanns, L. Thiele. "SPEA2: Improving the Strength Pareto Evolutionary Algorithm." *Technical Report 103, Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, Switzerland, May 2001. http://citeseer.ist.psu.edu/article/zitzler01spea.html*
129. K. Deb, M. Mohan, S. Mishra. "A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions". *KanGAL Report No. 2003002 (February, 2003) url: http://www.iitk.ac.in/kangal/papers/k2003002.pdf*
130. K.A. De Jong. "An Analysis of the Behaviour of a class of Genetic Adaptive Systems." *Ph.D. Thesis, University of Michigan, Dissertation Abstracts International 36(10), 5140B. 1975.*
131. H.P. Schwefel, "Numerical Optimization of Computer Models". *Chichester: Wiley & Sons, 1981.*
132. S. Langerman, "Definition of a test function for contest on Evolutionary Computation". *Proceedings of the Second IEEE Conference on Evolutionary Computation 1995, Piscataway, New Jersey, USA: IEEE Press, 1995.*
133. K. Deb. "Multi-Objective genetic algorithms: Problem difficulties and construction of test problems." *IEEE Evolutionary Computation Journal, 7(3):205-230, (1999)*
134. E. Zitler, K. Deb, L Thiele. "Comparison of multiobjective evolutionary algorithms: Empirical results", *Evolutionary Computation Journal, 8(2):125-148, (2000)*
135. K. Deb, L. Thiele, M. Laumanns, E Zitler. "Scalable Multi-Objective Optimization Test Problems". *Proceedings of the Congress on Evolutionary Computation (CEC-2002). pages 723-728, (2002)*
136. Kanpur Genetic Algorithms Laboratory (KanGAL) url: http://www.iitk.ac.in/kangal/soft.htm
137. Purshouse, R.C., Fleming, P.J., "Evolutionary Many-Objective Optimisation: An Exploratory Analysis", *Evolutionary Computation, 2003. CEC '03. The 2003 Congress on ,Volume: 3 , Dec. 8-12, 2003 Pages:2066 - 2073 IEEE 2003*
138. W.D. Wirth, "Radar techniques using array antennas", *Text Book, The Institution of Electrical Engineers, London, ISBN 0 85296 798 5, pages 93-95, (2001).*
139. R.J, Mailloux, "Array grating lobes due to periodic phase, amplitude , and time delay quantization", *IEEE AP-S Trans. Vol 32, No 12, Dec 1984, pp 1364-1368.*
140. U. Nickel, "Subarray configurations for digital beamforming with low sidelobes and adaptive interference suppression." *Proceedings of IEEE international conference on Radar, Alexandria, USA, pages 714-719, (1995).*
141. C. Tarran, M. Mitchell, R. Howard, "Wideband phased array radar with digital adaptive beamforming", *High Resolution Radar and Sonar (Ref. No. 1999/051),*

IEE Colloquium , Pages:1/1 - 1/7, (11 May 1999).

142. E.T. Bayliss, "Design of Monopulse Difference Patterns With Low Sidelobes", *Bell Syst. Tech. J., vol 47, pp. 623-650, (May-June 1968).*
143. H. Alnajjar, D.M. Wilkes, "Adapting the geometry of a sensor subarray", *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on , Volume: 4 , Pages:113 - 116 vol.4, 27-30 April 1993*
144. A.P. Goffer, M. Kam. "Design of Phased Arrays in Terms of Random Subarrays". *IEEE Transactions on Antennas and Propagation, Vol 42. No 6, June (1994).*
145. X. Zhiyong, B. Zheng, L. Guisheng. "A method of designing irregular subarray architectures for partially adaptive processing". *Radar, 1996. Proceedings., CIE International Conference of , 8-10 Oct. 1996 Pages:461 - 464*
146. J.A Smolko, "Optimization of pattern sidelobes in arrays with regular subarray architectures", *Antennas and Propagation Society International Symposium, 1998. IEEE , Volume: 2 , Pages:756 - 759 vol.2, 21-26 June 1998*
147. J. Wang, H, Israelsson, R.G. North, "Optimum subarray configuration using genetic algorithms", *Acoustics, Speech, and Signal Processing, 1998. ICASSP '98. Proceedings of the 1998 IEEE International Conference on , Volume: 4 , Pages:2129 - 2132 vol.4 (12-15 May 1998)*
148. F.Ares, S.R. Rengarajan, J.A. Rodriguez, E. Moreno. "Optimal Compromise Among Sum and Difference Patterns Through Sub-Arraying." *IEEE Antennas and Propagation Society, AP-S International Symposium (Digest), v 2, 1996, p 1142-1145 (1996)*
149. G. Golino, "A genetic algorithm for optimizing the segmentation in subarrays of planar array antenna radars with adaptive digital beamforming", *Phased Array Systems and Technology, 2003., IEEE International Symposium on , Pages:211 - 216, 14-17 Oct. 2003*
150. A. Jacomb-Hood, E. Lier, "Multibeam active phased arrays for communications satellites", *Microwave Magazine, IEEE , Volume: 1 , Issue: 4 , Pages:40 - 47, Dec. 2000*
151. D.A. McNamara, "Synthesis of sub-arrayed monopulse linear arrays through matching of independently optimum sum and difference excitations", *IEE Proceedings-Microwaves, Antennas and Propagation, IEE Proceedings H , Volume: 135 , Issue: 5 , Pages:293 - 296, Oct. 1988*
152. J. Mohamed, R. Holden, "Grating lobe Minimization in Sum and Difference Beam Patterns", *Antennas and Propagation Society International Symposium, 2003. IEEE , Volume: 1 , Pages:772 - 775 vol.1, 22-27 June 2003*
153. L. McKane, J. Kandel, "Microbiology: Essentials and Applications", *2nd ed. McGraw-Hill, Inc., New York. (1996)*
154. J. Thompson, "Phased array antenna for satellite communications to small tactical platforms", Military Satellite Communications (Ref. No. 2000/024), IEE Colloquium on, Pages:6/1 - 6/6, 6 June 2000
155 P. Moscato. "On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms." *Technical Report Technical Report C3P Report 826, Concurrent Computation Program, California Institute of Technology, 1989.*

Appendix A – Pareto Optimal Results

This appendix contains the results of the exhaustive searches detailed in Chapter Six. The two and three-objective test case problem results are presented.

f1 Mainlobe Gain +dBs	f2 Maximum Sidelobe Level –dBs
7.18	101.75
10.70	76.35
12.04	67.97
12.04	67.97
12.64	35.00
12.93	27.10
13.20	24.50
13.73	24.42
13.98	21.86
13.98	21.86
13.98	21.72
14.46	21.32
14.46	20.22
14.69	20.02
14.69	20.02
14.69	19.06
14.92	18.87
14.92	18.27
15.14	18.21
15.35	17.62
15.35	17.62
15.56	16.87
15.77	16.16
15.77	16.16
15.97	15.30
16.16	14.75
16.35	14.26
16.35	14.26
16.54	13.74
16.72	13.22
16.72	13.22
16.90	12.71
16.90	12.71

Table A-1 Pareto Optimal Solutions – Two Objective Exhaustive Search

Three objective Pareto Set:

f1 Mainlobe Gain +dBs	f2 1st Sidelobe Level -dBs	f3 Maximum Sidelobe Level -dBs
7.180	101.755	101.755
10.702	76.351	76.351
12.041	67.968	67.968
12.041	67.967	67.967
12.640	35.004	35.004
12.640	37.378	23.771
12.640	50.386	14.196
12.640	37.378	23.771
12.640	40.201	18.000
12.640	42.125	14.016
12.925	29.827	25.767
12.925	29.827	27.104
13.201	34.622	18.529
13.201	42.710	12.317
13.201	34.622	18.529
13.201	35.851	16.480
13.201	27.093	24.497
13.468	25.198	21.265
13.468	30.075	16.696
13.468	30.075	16.696
13.468	29.964	21.171
13.468	32.905	13.902
13.468	29.366	16.865
13.728	24.440	21.816
13.728	27.522	19.001
13.728	26.407	19.027
13.728	27.619	18.898
13.728	32.193	15.417
13.728	24.423	24.423
13.728	36.521	12.931
13.728	26.571	18.870
13.979	29.342	12.178
13.979	23.572	21.858
13.979	23.892	21.719
13.979	42.400	12.143
13.979	29.342	12.178
13.979	29.261	17.264
13.979	23.261	21.715
13.979	27.484	14.381
13.979	28.013	14.364
13.979	25.977	17.313
13.979	24.937	17.347
14.224	31.205	8.550
14.224	31.202	9.893
14.224	26.959	15.939
14.224	24.285	19.475
14.224	24.285	19.475
14.224	29.590	11.479

f1 Mainlobe Gain +dBs	f2 1st Sidelobe Level -dBs	f3 Maximum Sidelobe Level -dBs
14.224	29.590	11.479
14.224	23.987	15.972
14.224	23.262	16.008
14.224	22.457	19.377
14.224	23.847	15.997
14.224	23.262	16.008
14.224	26.959	15.939
14.224	29.251	13.426
14.224	23.987	15.972
14.224	22.382	19.578
14.462	27.291	10.903
14.462	21.317	21.317
14.462	26.081	12.654
14.462	20.225	20.225
14.462	26.794	10.901
14.462	23.250	17.779
14.462	23.040	17.796
14.462	26.794	10.901
14.462	21.049	17.879
14.462	25.088	14.851
14.462	21.253	17.828
14.462	26.969	8.197
14.462	21.156	17.862
14.694	23.736	10.383
14.694	20.709	20.019
14.694	20.709	20.019
14.694	21.715	16.453
14.694	22.947	13.930
14.694	22.947	13.930
14.694	21.981	16.447
14.694	23.693	11.979
14.694	23.693	11.979
14.694	21.981	16.447
14.694	19.926	16.468
14.694	19.061	19.061
14.919	21.278	13.140
14.919	18.869	18.869
14.919	21.711	11.383
14.919	19.874	18.307
14.919	20.707	15.344
14.919	19.764	18.272
15.139	19.854	12.450
15.139	19.470	14.409
15.139	19.892	12.450
15.139	19.031	16.940
15.139	18.720	16.941
15.139	19.031	16.940
15.139	19.470	14.409

f1 Mainlobe Gain +dBs	f2 1st Sidelobe Level -dBs	f3 Maximum Sidelobe Level -dBs
15.139	18.209	18.209
15.139	19.783	12.445
15.139	18.919	16.929
15.139	19.420	14.405
15.354	17.620	17.620
15.354	18.045	15.816
15.354	17.620	17.620
15.354	18.583	11.841
15.354	18.408	13.603
15.354	18.408	13.603
15.563	17.418	12.898
15.563	17.401	12.900
15.563	17.178	14.864
15.563	17.190	14.864
15.563	17.190	14.864
15.563	17.178	14.864
15.563	17.401	12.900
15.563	16.874	16.874
15.767	16.455	14.046
15.767	16.476	12.278
15.767	16.423	14.043
15.767	16.160	16.160
15.767	16.476	12.278
15.767	16.160	16.160
15.967	15.541	15.301
15.967	15.642	13.325
15.967	15.632	13.329
16.162	14.959	14.469
16.162	14.750	14.750
16.353	14.264	14.264
16.353	14.264	14.264
16.353	14.277	13.739
16.540	13.739	13.739
16.723	13.221	13.221
16.723	13.221	13.221
16.902	12.714	12.714
16.902	12.714	12.714

Table A-2 Pareto Optimal Solutions – Three-Objective Exhaustive Search

Appendix B – Definition of Antenna Terms

Definition of Standard Antenna Terms.

Term	Description
antenna	That part of a transmitting or receiving system that is designed to radiate or to receive electromagnetic waves.
adaptive antenna system.	An antenna system having circuit elements associated with its radiating elements such that one or more of the antenna properties are controlled by the received signal.
antenna array	*See:* **array antenna.**
aperture distribution.	The field over the aperture as described by amplitude, phase, and polarization distributions. *Syn:* **aperture illumination.**
aperture illumination.	*See:* **aperture distribution.**
array antenna.	An antenna comprised of a number of identical radiating elements in a regular arrangement and excited to obtain a prescribed radiation pattern. *Syn:* **antenna array.**
array element.	In an array antenna, a single radiating element or a convenient grouping of radiating elements that have fixed relative excitations.
array factor.	The radiation pattern of an array antenna when each array element is considered to radiate isotropically.
back lobe.	A radiation lobe whose axis makes an angle of approximately 180 degrees with respect to the beam axis of an antenna.
Bayliss distribution, linear.	A continuous distribution of a line source that yields a difference pattern with a sidelobe structure similar to that of a sum pattern produced by a Taylor linear distribution.
beam steering.	Changing the direction of the major lobe of a radiation pattern.
beamwidth. *See:*	**halfpower beamwidth.**
boresight.	*See:* **electrical boresight; reference boresight.**
boresight error.	The angular deviation of the electrical boresight of an antenna from its reference boresight.
conformal antenna [conformal array].	An antenna [an array] that conforms to a surface whose shape is determined by considerations other than electromagnetic; for example, aerodynamic or hydrodynamic.
difference pattern.	A radiation pattern characterized by a pair of mainlobes of opposite phase, separated by a single null, plus a family of sidelobes, the latter usually desired to be at a low level. *Contrast with:* **sum pattern.** NOTE—Antennas used in many radar applications are capable

	of producing a sum pattern and two orthogonal difference patterns. The difference patterns can be employed to determine the position of a target in a right/left and up/down sense by antenna pattern pointing, which places the target in the null between the twin lobes of each difference pattern.
Directivity (of an antenna in a given direction)	The ratio of the radiation intensity in a given direction from the antenna to the radiation intensity averaged over all directions.
Dolph-Chebyshev distribution.	A set of excitation coefficients for an equispaced linear array antenna such that the array factor can be expressed as a Chebyshev polynomial.
effective isotropically radiated power.	*See:* **equivalent isotropically radiated power.**
effective radiated power (ERP).	In a given direction, the relative gain of a transmitting antenna with respect to the maximum directivity of a half-wave dipole multiplied by the net power accepted by the antenna from the connected transmitter. *Contrast with:* **equivalent isotropically radiated power.** *Syn:* **equivalent radiated power.**
electronic scanning.	Scanning an antenna beam by electronic or electric means without moving parts. *Syn:* **inertialess scanning.**
equivalent isotropically radiated power (EIRP).	In a given direction, the gain of a transmitting antenna multiplied by the net power accepted by the antenna from the connected transmitter. *Syn:* **effective isotropically radiated power**.
excitation (of an array antenna).	For an array of radiating elements, the specification, in amplitude and phase, of either the voltage applied to each element or the input current to each element.
excitation coefficients.	The relative values, in amplitude and phase, of the excitation currents or voltages of the radiating elements of an array antenna.
far-field (radiation) pattern.	Any radiation pattern obtained in the far-field of an antenna.
gain (in a given direction).	The ratio of the radiation intensity, in a given direction, to the radiation intensity that would be obtained if the power accepted by the antenna were radiated isotropically.
grating lobe.	A lobe, other than the mainlobe, produced by an array antenna when the interelement spacing is sufficiently large to permit the in-phase addition of radiated fields in more than one direction.
half-power beamwidth.	In a radiation pattern cut containing the direction of the maximum of a lobe, the angle between the two directions in which the radiation intensity is one-half the maximum value.

	*See:***principal half-power beamwidths.**
inertialess scanning.	*See:* **electronic scanning.**
isotropic radiator.	A hypothetical, lossless antenna having equal radiation intensity in all directions.
linear array antenna.	A one-dimensional array of elements whose corresponding points lie along a straight line.
mainlobe.	*See:* **major lobe.**
major lobe.	The radiation lobe containing the direction of maximum radiation. *Syn:* **mainlobe.** NOTE—In certain antennas, such as multilobed or splitbeam antennas, there may exist more than one major lobe.
maximum relative sidelobe level.	*See:* **sidelobe level, maximum relative.**
mean sidelobe level.	The average value of the relative power pattern of an antenna taken over a specified angular region, which excludes the main beam, the power pattern being relative to the peak of the main beam.
minor lobe.	Any radiation lobe except a major lobe. *See:* **back lobe; sidelobe.**
monopulse.	Simultaneous lobing whereby direction-finding information is obtainable from a single pulse.
multi-beam antenna.	An antenna capable of creating a family of major lobes from a single non-moving aperture, through use of a multiport feed, with one-to-one correspondence between input ports and member lobes, the latter characterized by having unique main beam pointing directions. NOTE—Often, the multiple main beam angular positions are arranged to provide complete coverage of a solid angle region of space.
mutual coupling effect (A) (on the radiation pattern of an array antenna).	For array antennas, the change in antenna pattern from the case when a particular feeding structure is attached to the array and mutual impedances among elements are ignored in deducing the excitation to the case when the same feeding structure is attached to the array and mutual impedances among elements are included in deducing the excitation.
mutual coupling effect (B) (on input impedance of an array element).	For array antennas, the change in input impedance of an array element from the case when all other elements are present but open-circuited to the case when all other elements are present and excited.
null steering.	To control, usually electronically, the direction at which a directional null appears in the radiation pattern of an operational antenna.
pencil-beam antenna.	An antenna whose radiation pattern consists of a single mainlobe with narrow principal half-power beamwidths and

	sidelobes having relatively low levels. NOTE—The mainlobe usually has approximately elliptical contours of equal radiation intensity in the angular region around the peak of the mainlobe. This type of pattern is diffraction-limited in practice. It is often called a sum pattern in radar applications.
phase centre	The location of a point associated with an antenna such that, if it is taken as the phase centre of a sphere whose radius extends into the far field, the phase of a given field component over the surface of the radiation sphere is essentially constant, at least over that portion of the surface where the radiation is significant. NOTE 1- Some antennas do not have a unique phase centre.
phase pattern (of an antenna).	The spatial distribution of the relative phase of a field vector excited by an antenna. NOTES 1—The phase may be referred to any arbitrary reference. 2—The distribution of phase over any path, surface, or radiation pattern cut is also called a phase pattern.
planar array.	A two-dimensional array of elements whose corresponding points lie in a plane.
power pattern.	*See:* **radiation pattern.**
radiating element.	A basic subdivision of an antenna that in itself is capable of radiating or receiving radio waves. NOTE—Typical examples of a radiating element are a slot, horn, or dipole antenna.
radiation pattern.	The spatial distribution of a quantity that characterizes the electromagnetic field generated by an antenna. *Syn:* **antenna pattern.**
radiation pattern cut.	Any path on a surface over which a radiation pattern is obtained. NOTE—For far-field patterns, the surface is that of the radiation sphere. For this case, the path formed by the locus of points for which □is a specified constant and □is a variable is called a “conical cut.” The path formed by the locus of points for which □is a specified constant and □is a variable is called a “great circle cut.” The conical cut with □equal to 90° is also a great circle cut. A spiral path that begins at the north pole (□= 0°) and ends at the south pole (□= 180°) is called a “spiral cut.”
radiator.	Any antenna or radiating element that is a discrete physical and functional entity.
radome.	A cover, usually intended for protecting an antenna from the effects of its physical environment without degrading its electrical performance.
relative	*See:* **sidelobe level, relative.**

sidelobe level.	
scanning (of an antenna beam).	A repetitive motion given to the major lobe of an antenna.
self-impedance (of an array element).	The input impedance of a radiating element of an array antenna with all other elements in the array open-circuited.
shaped-beam antenna.	An antenna that is designed to have a prescribed pattern shape differing significantly from that obtained from a uniform-phase aperture of the same size.
shoulder lobe.	A radiation lobe that has merged with the major lobe, thus causing the major lobe to have a distortion that is shoulder-like in appearance when displayed graphically. *Syn:* **vestigial lobe.**
sidelobe.	A radiation lobe in any direction other than that of the major lobe.
sidelobe level, maximum relative.	The maximum relative directivity of the highest sidelobe with respect to the maximum directivity of the antenna.
sidelobe level, relative.	The maximum relative directivity of a sidelobe with respect to the maximum directivity of an antenna, usually expressed in decibels.
sidelobe suppression.	Any process, action, or adjustment to reduce the level of the sidelobes or to reduce the degradation of the intended antenna system performance resulting from the presence of sidelobes.
signal processing antenna system.	An antenna system having circuit elements associated with its radiating element(s) that perform functions such as multiplication, storage, correlation, and time modulation of the input signals.
space-tapered array antenna.	An array antenna whose radiation pattern is shaped by varying the density of driven radiating elements over the array surface. *Syn:*
steerable-beam antenna system.	An antenna with a non-moving aperture for which the direction of the major lobe can be changed by electronically altering the aperture excitation or by mechanically moving a feed of the antenna.
sum pattern.	A radiation pattern characterized by a single mainlobe whose cross section is essentially elliptical, and a family of sidelobes, the latter usually at a relatively low level.
Taylor distribution, linear.	A continuous distribution of a line source that is symmetric in amplitude, has a uniform progressive phase, and yields a pattern with a main beam plus sidelobes. The sidelobe structure is symmetrical, with a specified number of inner sidelobes at a quasi-uniform height, the remainder of the sidelobes decaying in height with their angular separation from the main beam. NOTE—Taylor distributions are often sampled to obtain the excitation for a planar array.
thinned array	An array antenna that contains substantially fewer driven

antenna.	radiating elements than a conventional uniformly spaced array with the same beamwidth having identical elements. Interelement spacings in the thinned array are chosen such that no large grating lobes are formed and sidelobes are minimized.
tracking	A motion given to the major lobe of an antenna with the intent that a selected moving target be contained within the major lobe.
two-dimensional scanning.	Scanning the beam of a directive antenna using two degrees of freedom to provide solid angle coverage.
uniform linear array.	A linear array of identically oriented and equally spaced radiating elements having equal current amplitudes and equal phase increments between excitation currents.

Appendix C – Author's Publications

Two published papers are presented:

[77]	D.W Ansell, E.J. Hughes, "Use of multi-objective genetic algorithms to optimise the excitation and subarray division of multifunction radar antennas" *IEE Multifunction Radar and Sonar Sensor Management Techniques (Ref. No.01/173), p 8/1-4. (November 2001).*
[78]	D.W Ansell, E.J. Hughes, "Using multi-objective genetic algorithms to optimise the subarray partitions of conformal array antennas" *Twelfth International Conference on Antennas and Propagation. ICAP 2003 (IEE Conf. Publ. No.491), pt. 1, p 151-5 vol.1, (April 2003).*

Use of Multi-Objective Genetic Algorithms to Optimise the Excitation and Subarray Division of Multifunction Radar Antennas.

Darren Ansell. (Associate Member IEE)
BAE SYSTEMS, Systems Engineering Product R&D Dept.
W328J, Warton, Preston,
Lancashire, PR41AX.

Dr Evan J.Hughes. (Associate Member IEE)
Cranfield University, Department of Aerospace, Power and Sensors, RMCS, Shrivenham,
Swindon, SN6 8LA.

Abstract:
In this paper, a method of applying multi-objective genetic algorithms (MOGAs) to optimise the excitation and subarray division of an active array antenna is presented. This enables beampatterns with multiple desired characteristics such as low sidelobes, maximum power efficiency and nulls in particular locations to be generated. The division of the array elements into subarrays is also optimised simultaneously by the MOGA.

Keywords: Multi-objective genetic algorithm, active array, subarray, antenna optimisation.

1. Introduction to GAs and MOGAs

Active electronically scanned array antennas are becoming commonplace in designs of radar systems. Arrays of several thousand radiating elements achieve power levels comparable with earlier single feed mechanically scanned systems. In order to steer the array and produce radiation patterns with desired characteristics, careful control of the array excitation is needed, particularly if the array is to be used in a multifunction manner where quite different beam patterns are required from the same array. This presents a complex non-linear optimisation problem.

The division of the array into a number of subarrays can simplify arrays and feed networks but raises the question of the optimum subarray divisions to make. The use of subarrays can also improve the signal to noise ratio and promote formation of sum and difference patterns [1,2].

There are many techniques for optimising the array excitations needed such as hill-climbing, and least squares methods[3] but they can be prone to convergence on local minima or maxima. They also require a good starting point in order to find the global optimum. They generally converge to a single point hence providing a single solution to the problem. If these techniques were applied to the subarray division optimisation problem, a single result would not permit any trade-off analysis to be performed after a single run of the algorithm.

The development of global search techniques such as simulated annealing and other evolution-based methods can improve the problem of local convergence because they conduct global searches of the design space.

One such method is the genetic algorithm (GA). The GA is based on Darwin's Theory of Evolution where 'populations' of solutions are evolved over a number of 'generations'.

The GA samples the search space stochastically and is far less likely to converge on non-global optima. For background material the interested reader is referred to [4] which contains several papers on introductory GA use.

The applications for GAs are widening as they become accepted as useful optimisation techniques. This is particularly true in the field of electromagnetics and antenna design.

The GA itself is generic and relies on a distinct 'fitness function' to calculate a measure of success of a solution during the optimisation process. For example, the fitness function may be monitoring radiation pattern sidelobe levels when the GA is optimising the excitation of elements in an active array. The fitness function is usually the most computationally expensive part of the algorithm. This is especially true in complex antenna optimisation code.

Simple GAs converge to a single solution. In problems where there are several, often conflicting objectives (true of many engineering problems), a multi-objective genetic algorithm (MOGA) can be used which evolves a set of solutions (the population) towards the Pareto-optimal front where trade-off analysis can be performed to select a suitable solution.

2. Literature

There are relatively few papers published in the literature concerning optimisation of subarrays

using GA techniques. Wang et.al[6], proposed a method for the optimisation of seismic array subarray configuration. In their paper, the SNR performance of a 20 element array with inter-element spacing of 2.5km was optimised using a simple GA. The algorithm formed subarrays by switching off certain elements in the array. The technique was not used to generate optimum weights or amplitude tapers for each of the subarrays but still obtained a 26% improvement in SNR using simple on/off excitations. Other authors have applied the GA to 'pre-formed' subarrays and found optimum excitation tapers to apply to the subarrays with encouraging results[1,2].

3. Aims & Test Case

There have been many papers written on the application of genetic algorithms to antenna array optimisation and also on generic multi-objective genetic algorithms. This paper attempts to bring the techniques together and apply MOGAs to simultaneously optimise the array excitations and the subarray division of a planar array antenna.

The algorithm is to be capable of optimising the subarray divisions, the number of subarrays and the excitations to apply to the subarrays.

4. Chromosome Encoding

The first step was to determine a suitable chromosome encoding scheme. The chromosome encoding scheme used to represent the subarrays and excitations has to be resistant to the genetic operators of crossovers and mutation, that is, it must produce valid chromosomes after these operations have occurred.

A suitable encoding scheme was implemented by using five different binary chromosomes. The five chromosomes are independent of each other and subject to separate crossover and mutation operations.

For each solution, each chromosome contains enough information to generate up to 50 subarrays (early runs of the algorithm showed that the maximum number of subarrays likely to be generated was 46). This allows the chromosome to be of fixed length and avoids the added complication of dealing with variable length chromosomes. The chromosomes contain a certain amount of redundant information, but this does not appear to slow down or degrade the optimisation process.

The information stored in the chromosomes is used to 'grow' a subarray from an initial start-point in the array. Therefore the chromosomes actually contain a series of choices as to which of the elements surrounding a chosen start point are to be included in the subarray.

When decoding the chromosomes, the subarrays are formed, one at a time, and a status flag updated to indicate which of the 400 array elements have been chosen as subarray members. The genes in the first chromosome, chromosome 1, provide the start-point information and when decoded, point to a position along a vector containing all remaining valid start-points.

Chromosomes 2 to 4 indicate which of the 48 elements surrounding the start point are potential members of the same subarray. Specifically, chromosome 2 indicates which of the 8 elements surrounding the start point are also potentially in the array. Chromosome 3 contains 16-bit genes relating to the elements which surround those selected by chromosome 2. Finally, chromosome 4 contains 24-bit genes that relate to the 24 elements around those selected by chromosome 3. Figure 1 illustrates the decoding of chromosome 2 into the elements surrounding the start point.

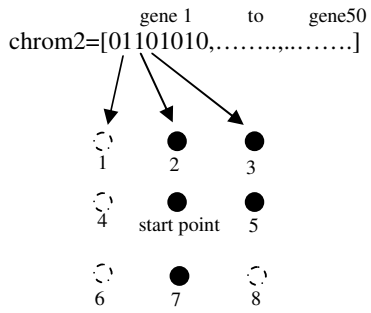


Figure 1 Chromosome Decoding Example.

At this stage, the subarray elements are only potential members of the subarray. The elements chosen by the chromosome only become valid subarray members after satisfying the following criteria:

- they are available (i.e. not members of other subarrays) and
- they are not isolated (i.e. they must be physically located next to other elements in the same subarray).

Once a subarray is formed, the status flag is updated and the procedure repeats until all the elements in the array become members of subarrays.

5. MOGAs and Pareto Ranking.

The solutions produced in each generation of the algorithm were ranked using the Niched Pareto Genetic Algorithm for Multiobjective Optimization proposed by Horn et al[5].

In the Niched Pareto algorithm, the solutions produced by the GA in each generation are ranked according to their dominance amongst the other solutions.

For example, if we consider a two objective problem, for a solution A to dominate solution B it must meet two criteria:

1. Each of the two objective values in A must be at least equal to each of the corresponding objective values in B
2. At least one objective in A must better the corresponding objective in B.

In the subarray problem, the objectives (or fitness values) (f_1, f_2, …f_n) relate to certain characteristics of the radiation pattern. For example, f_1 may measure maximum power output and f_2 maximum sidelobe level.

Niche sharing was used in order to distribute solutions along the Pareto front.

6. Array Synthesis

The array radiation pattern was calculated using the standard method contained within Skolnik[7] where the array factor of an arbitrary two-dimensional array is given by:

$$E_a(\cos\alpha_{xs}, \cos\alpha_{ys}) = \sum_{m=1}^{M} \sum_{n=1}^{N} \left|A_{(n,m)}\right| e^{j[m.(Tx-Txs)+n.(Ty-Tys)]} \quad \text{(Eq.1)}$$

where

$$T_{xs} = (\frac{2\pi}{\lambda}) d_x \cos\alpha_{xs} \quad \text{(Eq.2)} \qquad T_{ys} = (\frac{2\pi}{\lambda}) d_y \cos\alpha_{ys} \quad \text{(Eq.3)}$$

$$T_x = (\frac{2\pi}{\lambda}) d_x \cos\alpha_x \quad \text{(Eq.4)} \qquad T_y = (\frac{2\pi}{\lambda}) d_y \cos\alpha_y \quad \text{(Eq.5)}$$

The following equations were used to convert from a Cartesian to spherical coordinate system:

$$\cos\alpha_x = \sin\theta\cos\phi \quad \text{(Eq.6)} \qquad \cos\alpha_y = \sin\theta \ \sin\phi \quad \text{(Eq.7)}$$

$$\cos\alpha_{xs} = \sin\theta_s \cos\phi_s \quad \text{(Eq.8)} \qquad \cos\alpha_{ys} = \sin\theta_s \ \sin\phi_s \quad \text{(Eq.9)}$$

and $A_{(m,n)}$ = amplitude of the *mn*th element.

The test case for this algorithm was a 20 x 20 planar array of isotropic elements. The algorithm was set to produce 32 subarrays.

The algorithm was applied to the test case in order to produce a radiation pattern with low sidelobe levels. Half-wavelength element spacing was used.

7. Results

Early results are encouraging. After a run of just 30 generations with a population size of 30, the algorithm produced numerous solutions with maximum sidelobe levels less than 30dB. In a real problem, this would enable the antenna designer to perform trade-offs amongst these solutions. The design aim of 32 subarrays was successfully achieved. Figure 2 shows a typical antenna pattern selected at random from the 30 solutions in the final population provided by the GA. The multiple objectives measured included the maximum sidelobe level, the number of main beams produced and the total number of subarrays.

Future runs will use larger population sizes and more generations and make more use of the multi-objective capability to optimise beamwidth and power output levels. Figure 3 shows the subarray divisions provided by the algorithm and the normalised excitation values.

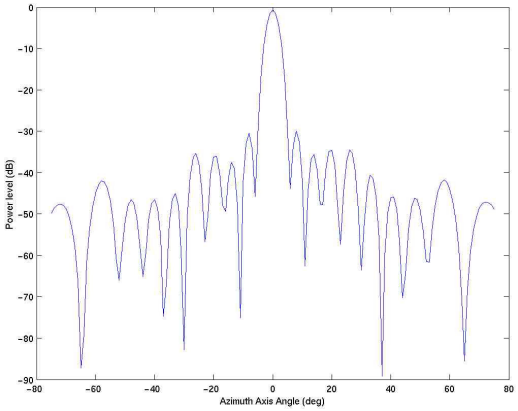


Figure 2. Sample Array Radiation Pattern

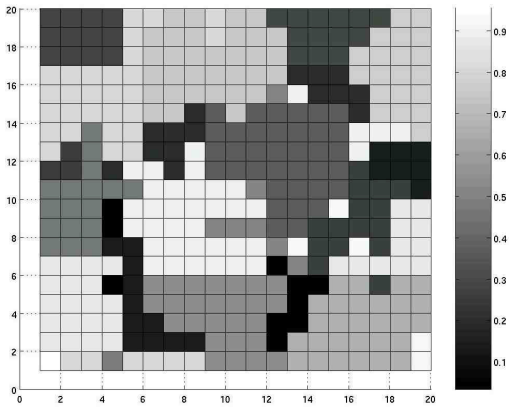


Figure 3. Subarray divisions and Excitation Values.

References

1. Optimization of Subarray Amplitude Tapers. Haupt, R. Antennas and Propagation Society International Symposium, 1995. AP-S. Digest , Volume: 4 , 1995. Page(s): 1830 -1833 vol.4

2. Optimal Compromise Among Sum and Difference Patterns Through Sub-arraying. Ares, F.; Rengarajan, S.R.; Rodriguez, J.A.; Moreno, E. Antennas and Propagation Society International Symposium, 1996. AP-S. Digest , Volume: 2 , 1996. Page(s): 1142 -1145 vol.2

3. Antenna Pattern Synthesis Using Weighted Least Squares. Carlson, B.D.; Willner, D. Microwaves, Antennas and Propagation, IEE Proceedings H , Volume: 139 Issue: 1 , Feb. 1992. Page(s): 11 –16

4. Electromagnetic Optimization by Genetic Algorithms. Rahmat-Samii, Michielssen, Wiley-Interscience, 1999. ISBN:0-471-29545-0

5. A Niched Pareto Genetic Algorithm for Multiobjective Optimization. Horn, J.; Nafpliotis, N.; Goldberg, D.E. Evolutionary Computation, 1994. IEEE World Congress on Computational Intelligence., Proceedings of the First IEEE Conference on , 1994 Page(s): 82 -87 vol.1

6. Optimum Subarray Configuration Using Genetic Algorithms. Wang, J.; Israelsson, H.; North, R.G. Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on , Volume: 4 , 1998. Page(s): 2129 -2132 vol.4

7. Radar Handbook, 2nd Ed. M Skolnik, McGraw-Hill, 1990. ISBN: 0-07-057913-X

USING MULTI-OBJECTIVE GENETIC ALGORITHMS TO OPTIMISE THE SUBARRAY PARTITIONS OF CONFORMAL ARRAY ANTENNAS.

D.W. Ansell[1], Dr. E.J. Hughes[2]
1, BAE Systems, UK, 2, Cranfield University, UK.

ABSTRACT

In this paper, a new and novel method for optimisation of subarray partitioning is presented that is inspired from a biological process. The process is optimised using a recent multi-objective genetic algorithm (MOGA) and demonstrated on a 15 x 15 element conformal array antenna. In installed arrays, element gain patterns are affected by mutual coupling and manufacturing tolerances and this optimisation technique takes these changes into account in order to optimise installed performance. Six objectives are used in the MOGA to optimise desired characteristics in the radiation pattern.

INTRODUCTION

Conformal array antennas are becoming increasingly important in the designs of future radar and communication systems, but do not enjoy the same coverage in the literature as planar array antennas.

The synthesis of conformal arrays is more complex than with planar arrays as the elements can all face in different directions. Therefore knowledge of the 'embedded' or 'active' element patterns must be considered when optimising operation and control variables. This is especially true in smaller arrays where the mutual coupling between elements is usually more significant. The mutual coupling coefficients can be estimated using a method of moments analysis, but it is better to use real measured patterns that can be stored and used in the optimisation process to improve accuracy of the results.

Once knowledge of the embedded patterns are known (or have been estimated), the complex excitations needed to steer the beam and reduce sidelobes can be optimised in order to give good installed performance. This is especially important in conformal arrays that generally exhibit higher sidelobes than planar arrays.

With a priori knowledge of the embedded patterns that include the affects of any EM coupling and radome distortion, it is possible to include them in an optimisation algorithm to improve the installed performance of an array. The problem is complex, with many hundreds of input variables (or thousands in large arrays) and multiple conflicting antenna performance objectives. The number of possible excitation sets is huge. One way to reduce the massive search space of possible excitation sets, is to partition the array into subarrays and apply a common excitation at the subarray level rather than at the element level. This raises the question of the best subarray partitions to use and the optimal excitation set to apply.

Genetic algorithms (GAs) are particularly well suited to complex problems of this type. This paper describes the application of Zitler's Strength Pareto Evolutionary Algorithm II (SPEA2) (1) to the optimisation of the complex excitations for a 225 element conformal array antenna (Fig. 1). A new and novel method for optimising the partitioning of conformal arrays into subarrays is also presented. The subarray partitioning is optimised simultaneously with the array excitations using SPEA2.

THE OPTIMISATION PROBLEM

In a standard or simple genetic algorithm (SGA), ‘populations’ of ‘individuals’ (solutions) are evolved in parallel over a number of ‘generations’ (iterations). An SGA samples the search space stochastically and is far less likely to converge on non-global optima than many of the classical optimisation techniques.

The SGA is generic and relies on a distinct ‘fitness function’ to calculate a measure of success for an individual during the optimisation process. In antenna optimisation, the fitness function may be monitoring sidelobe levels, beamwidth or gain. The fitness function is usually the most computationally expensive part of the algorithm as it involves synthesis and analysis of the antenna pattern. This is especially true when optimising large arrays.

When multiple objectives are to be optimised simultaneously, there is often more than one optimal answer. Solutions may be found that are good on one objective, but bad on another, forcing the designer to trade one objective for another. The set of optimal solutions is often called a trade-off surface, or more correctly, a Pareto optimal set.

SGA's converge to a point on the Pareto set and so provide a single optimised solution to a problem. This presents the system designer with little or no information on the shape of the Pareto trade-off surface unless multiple runs are of the optimiser are performed and the results are diverse (which is not usually the case).

When multiple objectives are evident in a problem, they must be combined in some way in a SGA in order to form a single objective. This is usually achieved by forming a (weighted) sum of objectives. When maximising a function, summing objectives can cause a SGA to oscillate around two or more maxima on the cost surface and leave concave regions of the Pareto set undisclosed.

Multi-objective Genetic Algorithms (MOGAs) however, carry only a marginal increase in processor overhead compared with SGA's and evolve a set of solutions to describe a diverse Pareto optimal set in objective space. In the case of antenna optimisation, the optimised trade-off surface delivered by the MOGA presents the designer with a choice of optimised operating points for the system. In theory, unlike a weighted SGA, all points on the surface described by the Pareto set could be found with a MOGA.

Antenna array excitation optimisation problems have very large search spaces due to the high numbers of elements and degrees (bits) of amplitude and phase control available. The use of subarrays reduces the number of the excitation values needed as a single excitation can be applied at the subarray rather than element level.

The method described below for finding optimal subarray partitions and excitation sets for good antenna array performance is one of effective genotypic representation in the evolutionary algorithm.

In a real array, in order to optimise beampattern performance to the highest possible degree, the subarray partitioning and excitations must be optimised simultaneously, taking into account the actual embedded gain patterns for each element. The method for performing this optimisation is described below and demonstrated on a conformal array antenna (Fig. 1).

LITERATURE

There are relatively few papers published concerning optimisation of subarrays using GA or EA techniques. Wang et al (2), proposed a GA method for the optimisation of seismic array subarray configuration. In their paper, the SNR performance of a 20 element array with inter-element spacing of 2.5km was optimised using a SGA. The algorithm formed simple subarrays by switching off certain elements in the array.

Haupt applied a GA to pre-defined subarrays and found optimum excitation tapers to apply to the subarrays with encouraging results (3). More recently, López et al (4) optimised linear array partitions and weights using a SGA.

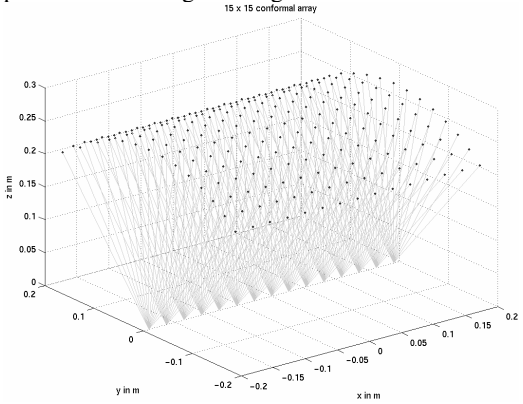


Figure 1. *Conformal Array Geometry. This is the testcase 225 (15 x 15) element array under optimisation. The lines projecting from the rear of the elements illustrate the pointing angle of each element. Array element spacing is 0.5λ in a 7 GHz system.*

SUBARRAY PARTITIONING METHOD

The method presented in this work for partitioning the antenna array into subarrays was inspired by the biological process of cell division. Single celled organisms such as amoebas, divide into two new cells by a process known as binary fission. The cells have to grow before they can divide, and so the speed with which they grow, to some extent determines their division rate. If we place together ***N*** live cells (each with a different growth rate) and observe their numbers over time, the two-dimensional

area occupied by each cell type will grow and form a 'footprint' below each cell type. This footprint will be unique in shape and area. The shape of the footprint is determined by the initial starting location of each 'seed' cell, the growth rate (cell division) rate for each cell and the initial position of each new cell formed. It is this biological process that has been modelled to represent the division of array elements into subarrays.

To demonstrate this method, consider a simple 9 x 9 element planar array. We initially form a 9 x 9 grid which represents the possible element locations in the array. The first step is to temporarily divide this grid into the required number of subarray partitions.

We then choose nine divisions and randomly place four seed 'cells' in each of the partitions (Fig. 2). The number of cells need not be fixed at four and can be varied by the user or by the optimisation process. The partitions are then discarded.

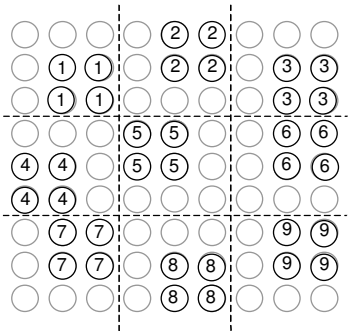


Figure 2. *Initial subarray seeds.*

We then randomly create an array of growth rate variables $\boldsymbol{R_{cell}}$, where $0 <= R_{cell} =< 1$. A final array $\boldsymbol{P_{cell}}$ is created in memory that acts as the trigger for cell reproduction. Initially $\boldsymbol{P}$ is set to zero and each value in $\boldsymbol{P}$ refers to the corresponding element located on the grid.

The pseudo code to describe this process is given by:

```
While array is not full:
  T = T + 1
  For all existing cells
    Pcell = Pcell + Rcell
    If Pcell > 1
      Form new cell of same type,
      placing new cell at nearest
      available orthogonal grid
      position.
    End If
  End For
End While
```

Each loop of the code represents one time period, T, during which the existing cells have time to 'grow'. Once they have reached a trigger size (as stored in $\boldsymbol{P}_{\text{cell}}$) they divide. Each new cell formed occupies the first available grid position in a north, west, east and then south order.

Although the grid is planar, the grid positions can easily represent conformal array element locations and so is suitable for both classes of antenna.

MULTI-OBJECTIVE SUBARRAY AND RADIATION PATTERN OPTIMISATION

To generate an entire array excitation set for the 15 x 15 array, including the 15 subarray partitions, 270 variables need to be optimised by the MOGA. This are broken down as:

- 15 x amplitude variables,
- 15 x initial seed location variables,
- 225 (15 x 15) x growth-rate variables.

In order to apply the MOGA, we have to represent a solution to our problem using a genotypic representation
of the variables (a chromosome encoding scheme).
The genotypic encoding schemes used to represent the above variables had to be resistant to the genetic operators of crossover and mutation; that means they must produce valid chromosomes after these operations have occurred.

Three real-valued chromosomes were used to represent a solution. The first ***ch1***, was used to represent the amplitude values. The second ***ch2*** encoded the initial seed locations and finally ***ch3*** contained the growth-rate variables.

GENERATION OF UNIQUE ELEMENT GAIN PATTERNS

For this theoretical study, unique randomised element gain patterns were generated and stored in a look-up table for use in the optimisation process. The gain patterns were generated by starting with a theoretical cosine pattern and then distorting at different angles by randomly attenuating the gain. The randomised plots were then 'smoothed' by calculating a polynomial approximation to each one. The polynomial approximation curve served as the unique gain patterns. When optimising the installed

performance of a real system, the gain patterns would first have to be measured or estimated for use in the optimisation process.

CONFORMAL ARRAY SYNTHESIS

In order to determine the success of a solution (defined by one set of chromosomes), it is necessary to model the far field pattern of the antenna. The far field radiation pattern of the conformal array was calculated using:

$$F(\theta,\phi)=\sum_{n=1}^{N} a_n \exp(j\alpha_n)\exp[jk(x_n\sin\theta\cos\phi+y_n\sin\theta\sin\phi+z_n\cos\theta)]FE_n(\theta_i,\phi_i)$$

(Eq. 1)

The steering phase required to steer the beam to θ_o,ϕ_o was calculated using:

$$\alpha_n=-k[x_n\sin\theta_o-\theta_n)\cos\phi_o-\phi_n)+y_n\sin\theta_o-\theta_n)\sin\phi_o-\phi_n)+z_n\cos\theta_o-\theta_n)]$$

(Eq. 2)

The multiple objectives used in the MOGA were chosen to characterise important defining features of the radiation pattern. The first fitness measure ***f1***, measured mainlobe gain and was encoded for maximisation, while ***f2*** measured the maximum sidelobe level (minimisation). The third, ***f3***, measured the maximum sidelobe level within ±40° of the mainlobe (minimisation). Both the azimuth and elevation beamwidths were measured and stored as ***f4*** and ***f5*** (both minimisation), with the magnitude of the difference between them being stored as another objective ***f6*** (minimisation) - this encouraged solutions to form with equal az. and el. beamwidths.

SPEA2 was chosen for this task as it is not limited by the number of objectives or input variables and in trials on this problem, has outperformed other recent MOGAs. A full explanation of SPEA2 is beyond the scope of this paper but in essence, SPEA2 uses the standard genetic algorithm operators of crossover and mutation to 'evolve' the input variables but maintains a fixed-size external archive of non-dominated solutions. It also includes measures to ensure the diversity in the non-dominated solutions found and maintained in the archive.

RESULTS

Several runs of SPEA2 were completed using 100 generations using the fitness measures defined above, population sizes of 100 individuals and a fixed external archive size of 35. Probability of crossover was 80 % and mutation 1 %. A single seed cell was used initially to grow each subarray. Once the maximum number of generations had been reached, the archive contained the best multi-objective solutions found during the run.

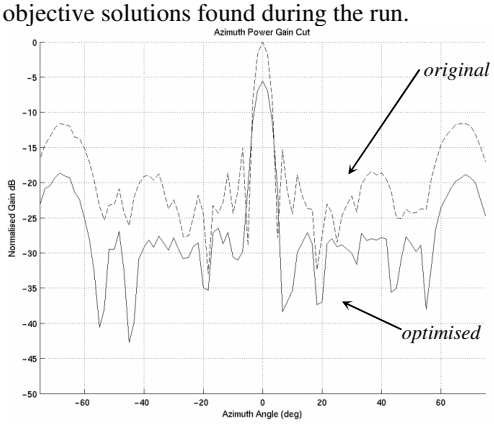


Figure 3. *Azimuth cuts of optimised (solid line) solution and a full power uniformly excited array (dashed line). Plot is normalised to 43.99dB (EIRP).*

Figures 3 and 4 show the az. and el. cuts of the radiation patterns of one such optimised solution contained in the archive set. Figure 5 shows the subarray partitions. Although the EIRP has dropped due to the tapering applied, the sidelobes have all been reduced in magnitude relative to the mainlobe, the az. and el. beamwidths are almost equal and the average sidelobe level is much lower than in full power transmission. With the exception of the far-out sidelobes, the sidelobe levels are relatively constant in the azimuth cut and low for an array of this size (15 x 15).

CONCLUSION

This work has demonstrated that state-of-the art multi-objective evolutionary algorithms can give good results when presented with difficult conformal antenna optimisation problems. In addition, a new and novel method for incorporating the subarray partitioning into the optimisation process has also been presented suitable for both planar and conformal antennas.

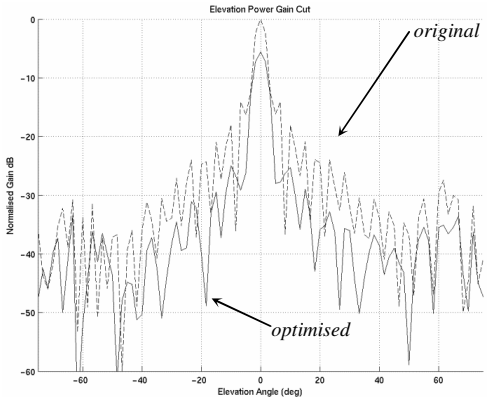


Figure 4. *Elevation cuts*

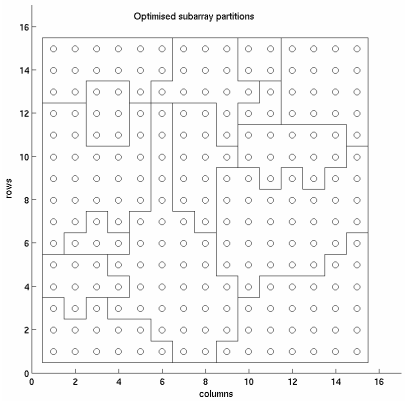


Figure 5. *Optimised Subarray partitions*

REFERENCES

(1) E. Zitler, M Laumanns, L Thiele (2001). SPEA2: Improving the Strength Pareto Evolutionary Algorithm. Technical Report 103, Computer Engineering & Networks Laboratory (TIK), ETH Zurich, Switzerland.

(2) J. Wang, H. Israelsson, R. G. North (1998). Optimum Subarray Configuration Using Genetic Algorithms. Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, Volume 4. pages 2129-2132.

(3) R. Haupt, (1995). Optimization of Subarray Amplitude Tapers. Antennas and Propagation Society International Symposium, AP-S. Digest, Vol.4, Page(s): 1830 -1833

(4) P. Lopez, A. Rodriguez. (2001). Subarray Weighting for the Difference Patterns of Monopulse Antennas: Joint Optimisation of Subarray Configurations and Weights IEEE Trans. on Antennas and Propagation, No 11 November 2001. Vol 49, Page(s): 1606 -1608